REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
20/Sep/2001	DISSERTATION	

4. TITLE AND SUBTITLE
THE AERIAL FLEET REFUELING PROBLEM

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
CAPT WILEY VICTOR D

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF TEXAS AUSTIN

8. PERFORMING ORGANIZATION REPORT NUMBER
CI01-256

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

13. SUPPLEMENTARY NOTES

14. ABSTRACT

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20011016 181

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		197	**19b. TELEPHONE NUMBER** *(Include area code)*

THE AERIAL FLEET REFUELING PROBLEM

by

VICTOR DUANE WILEY, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2001

The Dissertation Committee for Victor Duane Wiley
Certifies that this is the approved version of the following dissertation:

THE AERIAL FLEET REFUELING PROBLEM

Committee:

J. Wesley Barnes

J. Wesley Barnes, Supervisor

Paul A. Jensen

Paul A. Jensen

Leon S. Lasdon

Leon S. Lasdon

James T. Moore

James T. Moore

David P. Morton

David P. Morton

Valerie Tardif

Valerie Tardif

In love and gratefulness to my wife.

Acknowledgments

I would like to thank my wife for her constant love and support. Without her dedication and work at home with our boys, none of this would have been possible. I am thankful for her faith in me and her reminders that we serve God in all that we do.

I would like to thank my children for their patience and understanding during this time.

I would like to thank my advisor, Dr. Barnes, for his confidence in my ability. I appreciate the guidance he gave me during this endeavor and the freedom to explore different possibilities.

I would like to thank my committee for their suggestions and insight into the various ways to approach the AFRP.

I would like to thank my point of contact at AMC for the time he spent describing the AFRP and the data he provided.

THE AERIAL FLEET REFUELING PROBLEM

Publication No. ____________________

Victor Duane Wiley, Ph.D.
The University of Texas at Austin, 2001

Supervisor: J. Wesley Barnes

The deployment stage of the Aerial Fleet Refueling Problem (AFRP) for Air Mobility Command (AMC), Scott AFB, IL is efficiently solved using a Group Theoretic Tabu Search (GTTS). The GTTS uses the Symmetric Group on n-letters (S_n) and applies it to this problem using the Java^{TM} Object-Oriented Programming (OOP) language. The GTTS approach is sufficiently robust to be applied to other problem areas at AMC including the employment stage of the AFRP as well as the deployment and employment stages of the Airlift Problem. In the appendices, a brief description of the Java^{TM} implementation of the S_n, developed as an essential part of this research, is presented.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Air Refueling, the in-flight transfer of fuel between tanker and receiver aircraft, provides rapid response, increased range, and extended airborne operations for all aircraft. The capability to perform air refueling makes the United States (US) the dominant air power in the world today. This capability, coupled with the ability to efficiently employ air refueling assets, is essential for the US to maintain this dominance (AFDD 2-6.2 1999). The Air Mobility Command (AMC) of the United States Air Force (USAF) is responsible for determining the use of the US tanker fleet to meet the air refueling needs of the Air Force, Navy, Marines, US allies, and coalition partners. As part of this responsibility, AMC plans, schedules, tasks, and executes all operations involving the use of its forces. Part of AMC's planning encompasses the intertheater movement of forces from the US to areas around the globe. This "deployment" of forces and its accompanied air refueling requirement is known as the Aerial Fleet Refueling Problem (AFRP).

This chapter is organized as follows: Section 1.1 describes the basic AFRP and Section 1.2 outlines the major objectives of this research.

1.1 The Aerial Fleet Refueling Problem

1.1.1 Motivation

As the agency responsible for air refueling, AMC addresses many questions involving the allocation and use of tankers. During Operations Desert Shield and Desert Storm, approximately 400 tankers off-loaded over 1.2 billion pounds of fuel to over 80,000 aircraft while flying over 30,000 sorties and logging over 140,000 hours of flight time (*Gulf War Air Power Survey* 1993). The many planning scenarios, like Desert Shield/Desert Storm, that must be considered create complex sets of questions whose answers demand the use of powerful analytical tools. Among these tools are the Combined Mating and Ranging Planning System (CMARPS), the Quick-Look Tool (QLT) (Russina & Ruthsatz 1999), and the Tanker Assignment Planning (TAP) Tool (Capehart 2000).

CMARPS is a computer simulation that helps analyze, plan, and schedule deployment of tankers in support of immediate and anticipated military operations. Unfortunately, this tool can take up to two weeks to produce meaningful results. The QLT is a simple spreadsheet model used for determining gross tanker capabilities for supporting deployments. The TAP Tool is a spreadsheet model used for assigning tankers to refueling points. Their simplicity makes them incapable of addressing the problem in required levels of detail. AMC has an urgent need for a tool that will provide the detailed analysis of CMARPS within a planning horizon comparable to that required by the QLT. Given a deployment scenario, examples of overview questions that require answers are:

- How many tankers are required to meet the air refueling requirements?
- How quickly can all the receivers be deployed to their final destinations?
- How far do the tankers and receiver aircraft have to travel?
- How much fuel is burned by both tankers and receiver aircraft?

In order to meaningfully answer overview questions like these, a great many detailed questions, at the operational level, must be answered.

1.1.2 Problem Statement

We assume that the following information is given:

- a known set of tankers and their associated original beddown (starting) bases
- a known set of receiver aircraft, each with an initial departure base and a final arrival base, where one or more receiver aircraft are aggregated to form *Receiver Groups (RGs)*
- a known set of bases capable of refueling tankers and RGs
- a known set of flight characteristics for each aircraft including flight speed, altitude, take-off weight, fuel capacity, and fuel-burn rates
- a known set of tanker specific characteristics including fuel-offload capacity and fuel-offload rates

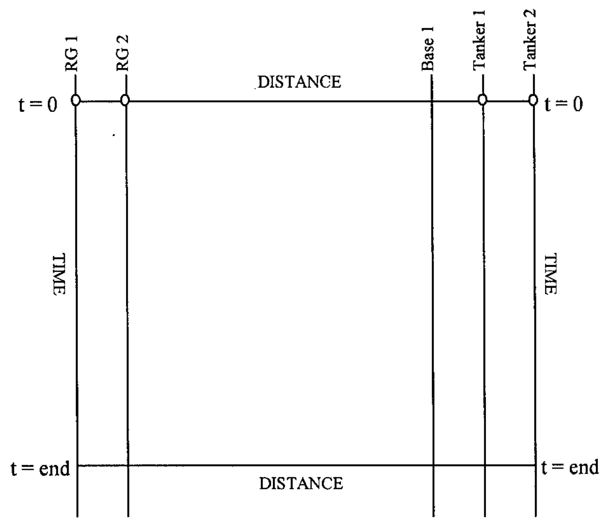

Figure 1.1: AFRP (time-distance view). This figure schematically presents an example of an AFRP scenario prior to any decisions. It provides each of the RGs, tankers, and arrival bases with an identifying label and provides a simplified frame of reference for elapsed time and distance traveled.

Using a "time-distance view" to represent the intersection of time and space (Yan & Tu 1997), Figure 1.1 displays an example of the information given by the first two bullets above. Similarly, Figure 1.2 provides a map-based view of Figure 1.1.

The complementary representations given in Figures 1.1 and 1.2 are typical of how an AFRP scenario is presented in this dissertation. The time-distance view associated with Figure 1.1 presents the operational details required by an analyst, in an easily understood schematic fashion, without the distractions of geographic detail. The map-based view provides the spatial and geographic information required by a decision maker to quickly assess such a scenario. Both views assist in the complete understanding of an AFRP

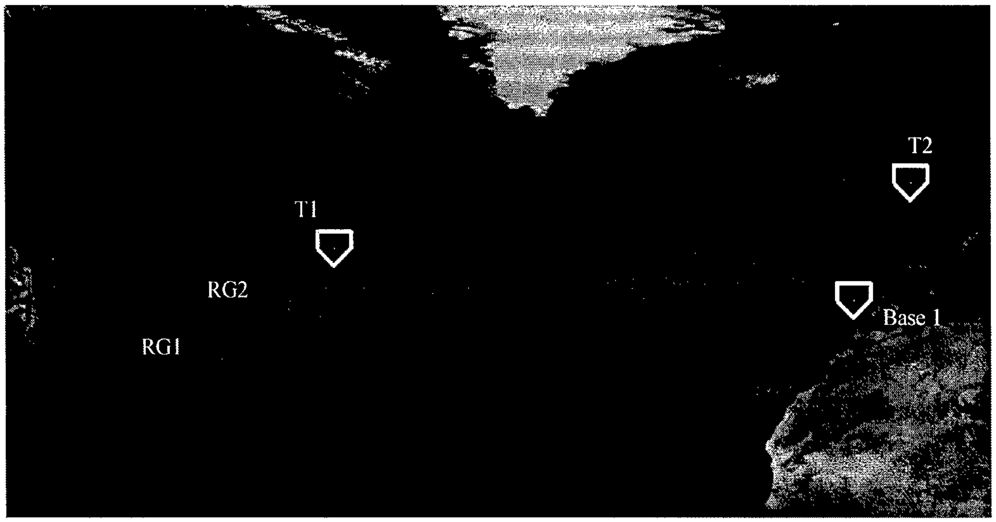

Figure 1.2: AFRP (map-based view). This figure geographically presents the AFRP pre-decision scenario of Figure 1.1. In this figure, the RGs' flight paths are displayed by the red lines. Each RG has two WPTs where refuelings are scheduled to take place before each RG arrives at a common destination base. The yellow pentagons represent the beddown bases of tankers 1 and 2.

scenario.

For a given deployment, the following *decisions* compose the *solution* to the AFRP

- the *waypoints (WPTs)*, i.e., the physical locations and start times where each refueling of all RGs will take place
- the tanker(s) that will serve each *WPT*
- how much fuel the assigned tanker(s) should deliver to a *WPT*

We further assume that the decision maker has the authority to (a) stipulate the departure times of all RGs and tankers and (b) vary flight speeds

(times) and/or require both tankers and RGs to "orbit" at specified locations to satisfy WPT requirements in terms of timing and location.

The AFRP's objective function is multicriteria and hierarchical in form. The hierarchical ordering of the associated criteria is subject to redefinition in accordance with perceived mission priorities. For the purposes of this research, the following criteria, in the order given, define the specific hierarchical objective function addressed:

Minimize:

1. the number of unescorted RGs requiring escort between WPTs
2. the number of WPTs not serviced by a tanker
3. the number of WPTs serviced out of an RG's flight path order
4. the amount of required RG fuel not supplied
5. the amount of time spent by RGs and tankers in "orbit" at a WPT(s)
6. the amount of RG late arrival time, i.e., where one or more RGs arrive later than a desired "soft" arrival time
7. the number of tankers used
8. the amount of tanker flight time required
9. the total distance flown by tankers
10. the amount of fuel used by tankers
11. the amount of fuel off-loaded by tankers

12. the amount of fuel used by RGs

These criteria are interpreted in a strict hierarchical fashion. Suppose that we are comparing two solutions, A and B. We first compare the values of criterion 1. If the criterion 1 values for solutions A and B are not identical, the solution with the lesser value is considered to be superior, regardless of the values of the other 11 criteria. If the two criterion 1 values are identical, the comparison procedure moves to criterion 2 where the same process is repeated. This sequence is repeated until superiority is determined or until all 12 criteria for solutions A and B are found to be identical. If, and only if, the latter case occurs, will solutions A and B be deemed equivalent.

The criteria are ordered as above for the following reasons:

Nonzero values of any of criteria 1 through 4 mark a solution as infeasible because of violation of USAF policy or because one or more aircraft is unable to complete their required flights. Criteria 5 thorough 12 exist and are present in the order given above due to explicit stipulations from AMC.

Figures ?? and 1.4 display a specific solution to the scenario of Figures 1.1 and 1.2. In this solution, Tanker1 services RG2 at WPT3 and returns to its beddown base. It then continues operations by flying to WPT1, servicing RG1, and then returning once again to its beddown base. Tanker2 services RG2 at WPT4 then flies to WPT2 and services RG1 before returning to its beddown base. In this solution, the assignment of tankers to WPTs influences the timing of other physical activities including:

- the departure time of each RG

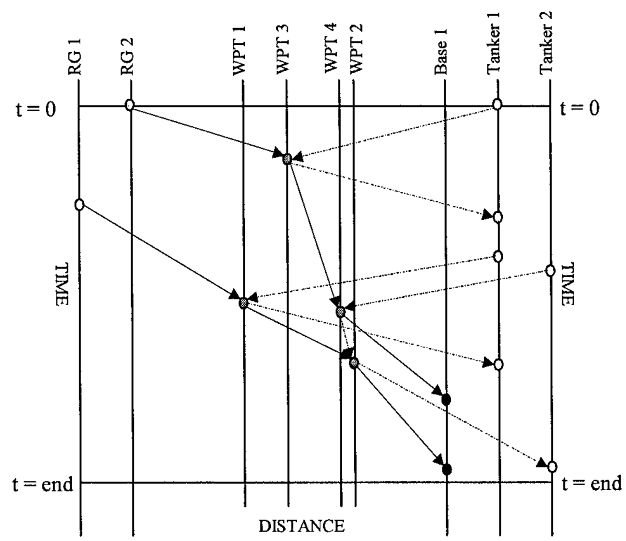

Figure 1.3: AFRP Solution (time-distance view). This figure depicts a specific solution to the example in Figure 1.1. The nodes along the vertical axis of RG1 and RG2 represent the departure times of the RG from their initial locations. The nodes along the vertical axis of Tanker1 and Tanker2 represent the initial and subsequent departure and arrival times of the tankers. The nodes along the vertical axis of the WPTs represent the service start times of the respective RG-tanker combinations as well as the actual service times. The nodes along the vertical axis of Base1 represent the RG arrival times.

- the initial and subsequent departure times of the tankers during their service to the deployment

Figure **??** clearly presents the solution from the viewpoint of the specific timing and time ordering of the solution events and the specific actors in each event (i.e., the RG and tanker at each WPT and the RG or tanker associated with each takeoff and landing). Figure 1.4 provides a map-based view of Figure **??**'s solution and clarifies the spatial interactions and locations of each solution event. The RGs' nonstationary character is a complicating and very important aspect of AFRP because it requires not only that a WPT's start

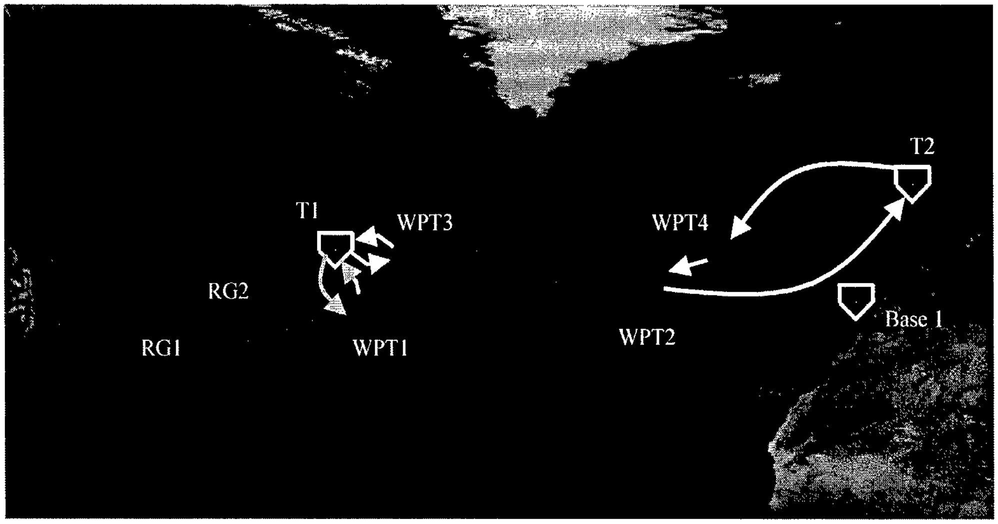

Figure 1.4: AFRP Solution (map-based view). This figure complements the solution presented in Figure ?? and elucidates the spatial interactions and locations of each solution event. Tanker1's first trip is in yellow and its second trip in green.

time be stipulated but also that the WPT's three dimensional location be stipulated. This special consideration in terms of the timing and location of events is not common in other routing problems. This is discussed in detail later.

In addition, the amount of fuel off-loaded at a WPT markedly affects such things as:

- the refueling service time at the WPT
- the spatial locations and start times of subsequent WPTs along the flight path of an RG

Thus, the solution to the AFRP is composed of a complex set of inter-related decisions involving time, space, and amounts of fuel. Usually, the effect

of changing any individual decision will "ripple" through the AFRP structure forcing multiple associated changes in the solution.

Fighter aircraft require continuous escort (*deployment support*) while flying over large bodies of water. The added consideration of deployment support restricts the assignment of tankers to WPTs by forcing a tanker to follow a RG along the RG's flight path. Tankers that provide escort typically do not immediately return to their original base. Instead, they often continue operations from another tanker base. Figures 1.5 and 1.6 provide an example of RG1 and RG2 being escorted during part of their flight to Base1. Tanker1 escorts RG1 from WPT1 to WPT2 and lands at Base1 while Tanker2 escorts RG2 from WPT3 to WPT4 and then returns to its beddown base. In this example, Tanker1 is serviced at Base1 and then returns to its beddown base. In other scenarios, Tanker1 could have been refueled at Base1 and continued to service the deployment without returning to its beddown base. (Note that these examples are for illustrative purposes only and may not accurately reflect reality.)

While a less common method, requiring more tankers and more tanker flight time, tankers may also receive midair refueling from other tankers subsequent to servicing additional RGs. A tanker that maintains its position in space while providing fuel and receiving fuel serves as a *temporary base.* This *air bridge support* may occur at a number of locations to provide continuous service along a given path. Figure 1.7 provides a time-distance view of air bridge support. In this figure, each tanker flies to its assigned spatial coordinate (labeled as LOC1 and LOC2) and waits for RGs to show up for service.

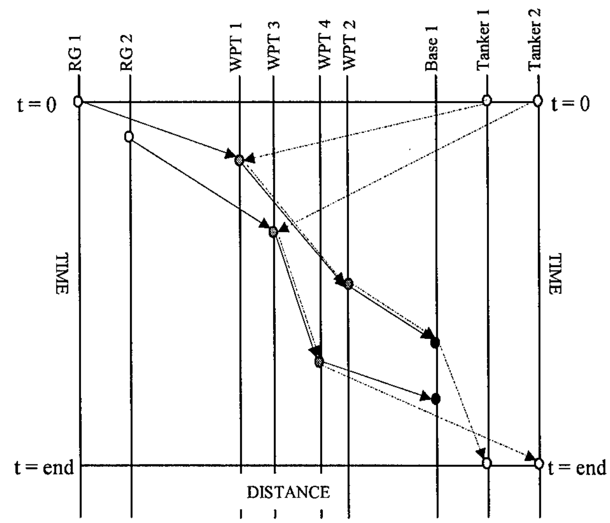

Figure 1.5: AFRP Escort (time-distance view). This figure depicts another solution to our example AFRP where escorts must be provided. In this solution, each tanker, after providing service at the initial WPT of a RG, escorts the RG to its next WPT. The nodes along the vertical axis of Base1 represent the arrival times of the RGs and Tanker1 to Base1. For Tanker1, its service time at Base1 is included as part of the node.

In this example, we assume that the two tankers have the capability to perform the refueling at LOC1 and LOC2 without additional support. In most air bridge support scenarios, employment of additional tankers is required. Figure 1.8 complements Figure 1.7 by showing how the RGs converge to LOC1 and then follow the same path to LOC2 and on to Base1. In this figure, RG1 and RG2 are serviced at LOC1 (WPT1 and WPT3, respectively) and at LOC2 (WPT2 and WPT4, respectively).

The AFRP solution is constrained by a large number of limiting factors. Primary among these constraints is the safety of the crews and aircraft associated with the deployment, i.e., no tanker or receiver aircraft should have

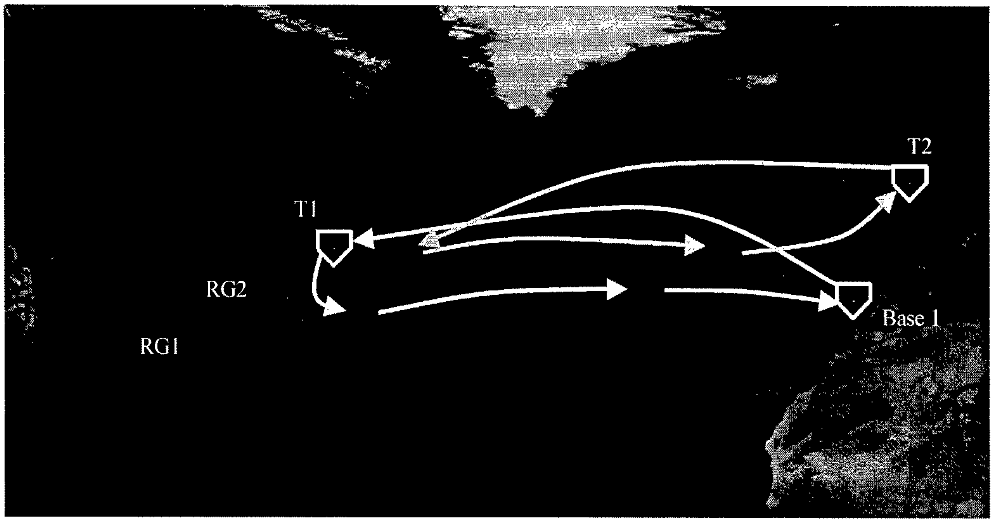

Figure 1.6: AFRP Escort (map-based view). This figure complements Figure 1.5 and clarifies the spatial impact of requiring deployment support, i.e., significantly greater travel by the tankers is required.

to divert from its flight path for lack of fuel. Many other constraints must also be satisfied. For example, a tanker has limited fuel capacity and its crew has flight duration restrictions which affect the crew-tanker ability to travel long distances and to provide fuel. Certain bases have limited capacity for resident tanker aircraft (known as *maximum on ground*, or MOG) . A tanker requires time to fly between any two locations and time to perform midair refueling. Hence, all tanker WPT assignments must be limited to those where the tanker is physically capable of being present at the specified WPT time.

1.1.3 Modified Formulations

The AFRP is unique, complicated, and extremely difficult to model and solve when viewed in its full practical context. However, by relaxing selected constraints and/or by restricting (i.e., fixing) selected decision variables, it can be

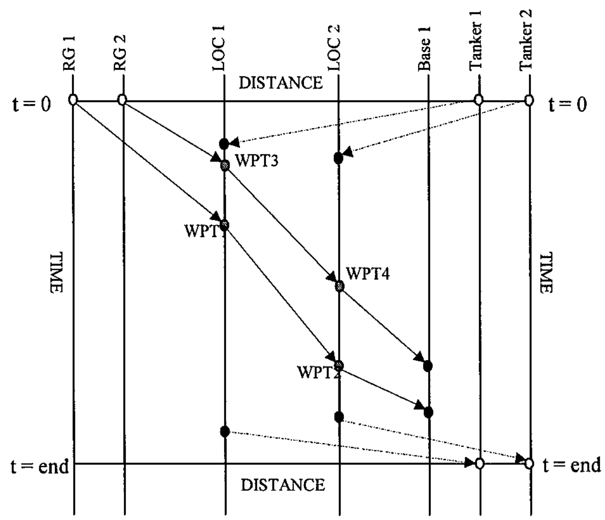

Figure 1.7: AFRP Air Bridge Example (time-distance view). This figure depicts an AFRP solution where two tankers provide air bridge support. Tanker1 flies to LOC1 where it "orbits" while waiting for RGs to arrive. Similarly, Tanker2 flies to LOC2 where it also waits. The tankers orbiting at LOC1 and LOC2 provide service to both RG1 and RG2 before returning to their respective beddown bases.

simplified to correspond to one of several classical mathematical programming problems.

Modeling the AFRP as a Vehicle Routing Problem (VRP)

The AFRP may be characterized, in the classification scheme of Barnes & Carlton (1996), as a variation on a type of Multi-Vehicle, Multi-Depot, VRP. In the AFRP, we have finite capacity heterogeneous vehicles (with route length/route duration upper bounds) that are required to deliver product to customers (such deliveries requiring a finite [> 0] service time).

In the notation of Barnes & Carlton (1996), the AFRP is a variation

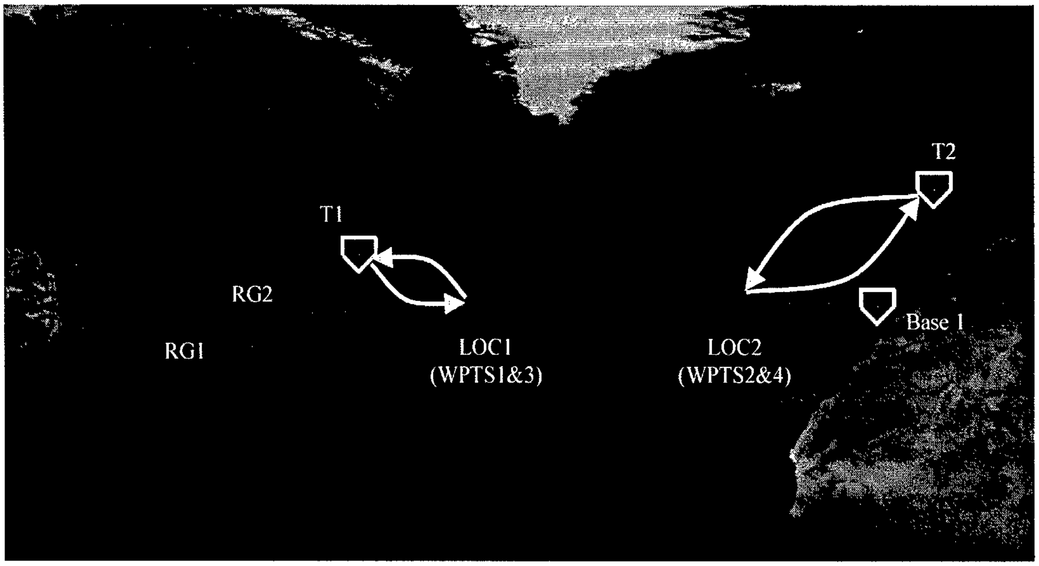

Figure 1.8: AFRP Air Bridge (map-based view). This figure gives the map-based view of the solution presented in Figure 1.7.

of problem type α which is known to be NP-hard (Carlton 1995, Gendreau, Laporte & Potvin 1997).

$$\alpha = (MV\overline{H}, MD, VRP, RL)$$

where $MV\overline{H} \equiv$ Multi-Vehicle Heterogeneous, $MD \equiv$ Multi-Depot, $VRP \equiv$ Vehicle Routing Problem, and $RL \equiv$ Route Length.

However, there are several additional considerations present in the AFRP that are not present in problem α. These are :

1. In problem α, the customers' locations are fixed in space, requiring only that a decision on the ordering (relative to other customers assigned to a vehicle) of the delivery and the responsible vehicle be stipulated. Further, the amount of product to be delivered to each customer is an *a*

priori stipulated deterministic amount and there is a single delivery to any customer. Finally, the route length restriction is given only in terms of a total travel distance that may not be exceeded. Problem α has no explicit accounting for the timing of events. In the AFRP, we know only the total amount of fuel that must be provided to an RG during its total travel. As in problem α, we must stipulate the responsible vehicle (tanker) and the ordering of any delivery. In addition, for all RGs, we must also stipulate the spatial location (longitude, latitude, and altitude in 3-dimensional space) and start time of each fuel delivery and the number of possibly multiple deliveries and the amount of product (fuel) to be provided in each delivery.

2. All customers must be supplied with fuel in a timely manner that will assure that no receiving aircraft has its available fuel fall below a prespecified "minimal reserve."

3. Directly associated with the WPT decisions are the decisions on the takeoff time of each RG and the possibly multiple takeoff times of each tanker.

If we desired to solve the AFRP by forcing it to be equivalent to problem α, all considerations of event timing must be relaxed; the spatial location and fuel requirement of each WPT must be fixed known constants; and each WPT becomes equivalent to a separate customer in the problem α formulation. Further, if the AFRP route-length restriction were given in terms of the maximal amount of time flown, this restriction would have to be converted to an equivalent distance flown restriction.

Problem α may be brought closer, in some respects, to the AFRP through the inclusion of a "time window" constraint on each of the WPTs, i.e., the stipulation of an earliest and latest time that refueling can begin at each WPT (customer). These additional constraints would yield another of the problem types described by Barnes & Carlton (1996):

$$\beta = (MV\overline{H}, MD, VRP, RL, TW)$$

using the notation of α with $TW \equiv$ Time Windows

The stipulation of time windows on the WPT refuelings would reintroduce some of the time-based considerations, i.e., time ordered precedence relations between events, but would still require the spatial location and fuel requirement of each WPT to be fixed known constants.

Modeling the AFRP as a Job Shop Scheduling Problem

The classical Job Shop Scheduling Problem (JSSP) (Baker 1974, French 1982) involves a stipulated number of jobs, n, each composed of a set of m technologically ordered operations. Each operation is assigned to one of a known set of m machines and a machine performs one operation for any job. Each operation requires a known amount of time for completion and the objective is to minimize the time required to complete all jobs. There are a total of $(n!)^m$ possible orderings of the operations on the machines for the n-job, m-machine JSSP. The JSSP is known to be NP-hard (Garey, Johnson & Sethi 1976).

The AFRP can be viewed as a generalization of the JSSP in the following way. Let each RG be a "job" and let the set of WPTs assigned to the RG be the technologically (temporally) ordered "operations" associated with

that job. The tankers may be viewed as the JSSP "machines." The differences between the JSSP and the AFRP are:

1. tankers can service any subset of WPTs and the WPT assignment to tankers is not stipulated, but rather determined as part of the solution process,

2. there is a sequence dependent set-up time *and* cost associated with the assignment of any WPT to any tanker, i.e., the time required for the tanker to fly to the WPT and the fuel used during that flight,

3. the number of tankers and WPTs are not stipulated and are determined by the solution process, and

4. the objective function for the AFRP is multicriteria and hierarchical rather than unidimensional for the JSSP.

Modeling the AFRP as a Set Covering Problem

A Set-Covering Problem (SCP) is an NP-hard problem (Papadimitriou & Steiglitz 1982) consisting of a finite set $M = \{1, \ldots, m\}$ from which a collection of subsets exists $\{M_j\}$ for $j \in N = \{1, \ldots, n\}$ with an associated cost, c_j, for each. A subset $F \subseteq N$ covers M if $\cup_{j \in F} M_j = M$. The SCP seeks to find the minimum-cost cover from $\mathcal{F} = \{F : F \text{ covers } M\}$ (Nemhauser & Wolsey 1988).

If accounting for the timing of events is relaxed and the decisions of the WPTs' physical *locations*, fuel *requirements*, **and** *all* feasible routes for *all* tankers are given, the AFRP reduces to a SCP. In this context, all such feasible routes would explicitly satisfy all necessary timing and route-length

restrictions associated with the problem. The emphasis of this formulation is to determine the best set of feasible routes to assign to the tankers. In this sense, the SCP can be viewed as a vehicle routing problem of type α with the decisions of which customers to serve with each tanker aggregated into all possible feasible sets.

Modeling the AFRP as an Assignment Problem

The Assignment Problem (AP) as defined by Nemhauser & Wolsey (1988) concerns n servers and m jobs, where $n \geq m$. Each job must be done by exactly one server; also, each server can perform, at most, one job. The cost of server j performing job i is c_{ij}. The problem is to assign servers to the jobs so as to minimize the total cost of completing all of the jobs.

If accounting for the timing of events is relaxed and the decisions of the WPTs' physical *locations*, fuel *requirements*, **and** *feasible* routes (satisfying all AFRP constraints) for all tankers are given where the routes contain exactly one WPT each, then the AFRP reduces to an AP. The AFRP may have many more WPTs than tankers, so the reuse of tankers would be required to satisfy the $n \geq m$ condition. The emphasis of this formulation is to determine the best set of routes (assignments) for all single WPT tanker trips.

The above discussion shows that problems α and β, the SCP, and the AP may be viewed as special cases of the AFRP. Furthermore, the AFRP is a generalization of problems α and β, in the dimensions of space (customer locations) and demand (fuel requirements). Thus, the AFRP is NP-hard because it is a generalization of problems known to be NP-hard.

1.2 Research Objectives

The primary objectives of this research were:

1. to establish a foundation and "proof of concept" for the heuristic solution of the AFRP and

2. to develop methods for producing a "suite" of excellent solutions to any instance of the AFRP.

Upon the foundation of objective 1, future enhancements and additional detail can and will be built by future researchers. To ensure that these expansions can take place, the foundation is formulated within a reusable, portable framework.

To address objective 2, a Group Theoretic Tabu Search (GTTS) approach was used. The GTTS makes use of adaptive tabu search to dynamically update the memory structures as well as to promote diversification. The GTTS casts a solution to the AFRP as an element from the Symmetric Group on n-letters (S_n) and creates move neighborhoods that can be represented by the Symmetric Group acting under conjugation or multiplication upon itself. To make this possible, a Java™ class library for S_n was created by the author. This class library is described in Appendices C and D. To address the issue of reusability and portability, the GTTS approach is implemented using the Java™ programming language. Secondary research goals were to investigate the effects of selecting different move neighborhoods both from a static and dynamic context. How these goals were accomplished is discussed in detail in the remainder of this dissertation.

Chapter 2

Literature Review and Mathematical Foundations

The idea of in-flight refueling has existed almost as long as the airplane itself. The efficient performance of aerial fleet refueling is a much more recent concept. Prior to the developments documented in this dissertation, only limited tools were available to assist a decision maker in the accomplishment of that task.

This chapter is organized as follows: Section 2.1.1 provides a brief history of aerial refueling within the US and discusses previous research specifically directed at the AFRP. Section 2.2 briefly recounts directly relevant literature underlying the four classical formulations, the VRP, JSSP, SCP, and AP (discussed in Chapter 1), that result from relaxed or restricted forms of the AFRP. Section 2.3 overviews the algebraic foundations essential to the understanding of the group theoretic concepts that are utilized in the rest of this dissertation.

2.1 Aerial Refueling

2.1.1 Brief History

The concept of aerial refueling can be traced back to 1918 (Strategic Air Command 1990). In 1921, Sunderman (1961) reports that a man got out of the open cockpit of a Lincoln Standard biplane and physically delivered five gallons of fuel to a JN-4 receiver aircraft. Within the US, the first refueling using a hose to connect the aircraft took place on June 27, 1923 between two Army Air Service airplanes. The historic flight of the *Question Mark* (a modified Atlantic (Fokker) C-2A) on January 1, 1929 demonstrated aerial refueling's military potential by establishing a world duration record of 150 hours and 40 minutes by utilizing 43 refuelings from two modified Douglas C-1 biplanes. The flight was prematurely terminated due to engine problems on the *Question Mark.* After this flight, aerial refueling received some attention, but the advent of more efficient aircraft designs stalled its progress. It was not until the creation of the USAF and the need for airplanes to reach the USSR (due to the Berlin Blockade) that the US officially incorporated the idea of aerial refueling. This occurred in 1948 with the formation of two refueling squadrons. The Cold War caused the need for refueling of our strategic bomber force and, subsequently, the requirement for aerial refueling has encompassed all aspects of the USAF mission from tactical fighters to logistic airlifters. For a complete historical review of aerial refueling in the USAF, the reader is referred to Strategic Air Command (1990), Byrd (1994), Moncrief (1996) and Smith (1998).

2.1.2 Efficiency Research

Currently, the demand for aerial refueling can often exceed the USAF's ability to satisfy that demand in a desired period of time. The need for an efficient and effective planning tool for the AFRP has long been acknowledged.

Presently maintained for AMC by LOGICON, Inc. (a subsidiary of Northrup Grumman), the first planning tool, CMARPS, was developed about twenty years ago (Ryer 2001). CMARPS is a deterministic computer simulation (emulation) program that helps analyze, plan, and schedule deployment of tankers in support of immediate and anticipated military operations. This *legacy* program was designed to assist in the planning, allocation, and scheduling of air refueling assets during peacetime, crisis, contingency, and wartime. Unfortunately, because of the great number of scenarios that must be explicitly constructed, this tool can take up to two weeks to produce meaningful results. An extremely large multivolume user's manual is available only in hardcopy and provides no detail on the methodology employed by CMARPS.

While CMARPS can assist in providing extensive, detailed, and accurate data for predicting receiver and tanker aircraft mission requirements, CMARPS' marked complexities make quick and effective use difficult even for *highly* experienced users. In addition, the extensive computing power requirements limit CMARPS efficiency, mobility, and versatility. CMARPS currently consists of four main components (Ryer 2001):

1. The Combined Mating and Ranging Planning System (CMARPS) determines refueling support for a *single* Receiver Group (RG) by assisting in the determination of refueling locations that attempt to minimize the use of

tanker air frames and flight times. The program determines tanker capabilities and assigns tanker support after actually performing flight emulation for both the receiver and the tanker aircraft. CMARPS generates flight plans for receivers and tankers while generating summaries for solution analysis.

2. The Graphically Supported Interactive Control System (GSICS) provides the graphical user interface that allows the planner to modify program input to support the other components while providing a Mapping, Charting, Geodesy, and Imagery (MCGI) capability.

3. The Tanker Mating and Ranging Program (TMARP) determines refueling support for multiple RGs and schedules RG movement according to tanker and crew availability. There are three major components of TMARP: Deployment, Employment, and Horseblanket (an online capability for tanker units to "buy and sell" air refueling missions in support of the quarterly tanker allocation scheduling process).

4. TPFDD Sizing, Sourcing, and Analysis System (TSSAS) supports Time Phased Force Deployment Document (TPFDD) analysis, updating, and modification and provides the capability to determine and source AMC deployment flow support requirements.

The limitations of CMARPS in terms of its computational inefficiency, extreme complexity, and intense requirement for analyst interaction have brought forth several later studies on ways to improve on these limitations. Initial attempts at this strategic level problem include Yamani (1986) and Hostler (1987). Yamani (1986) addressed not only aerial refueling concerns but also the questions of how many aircraft would be needed in a receiver group and

how each aircraft would be loaded. His research associated with the AFRP was limited to two cases: (a) consideration of RGs where only one waypoint (WPT) was required and (b) consideration of single aircraft that required two WPTs. Hostler (1987) allowed a tanker to refuel at most two WPTs and ignored the travel distance and fuel consumption limitations associated with the tankers.

More recent research includes Russina & Ruthsatz (1999) and Capehart (2000). Russina & Ruthsatz (1999)' QLT was an initial attempt to provide AMC with a more responsive program than CMARPS. The primary goals of this augmented Microsoft Excel spreadsheet tool were to determine a feasible number of tankers needed for a planned deployment and then determine how quickly that deployment could be achieved. Several simplifying assumptions were incorporated in the QLT analysis structure. Among these were (a) constant flight speeds for all aircraft, (b) the refueling tanker would also provide any escort duties required, (c) all tankers were identical, (d) only one tanker could be assigned to a refueling point regardless of the amount of fuel required, (e) after completing its duties, which may contain at most one refueling, each tanker must return to its home base, and (f) the locations of all WPTs and the amount of fuel required at the WPT are assumed to be known constants and are part of the input data. The most limiting assumption in the QLT was that, when escort was not required, all tankers were assumed to take the same constant amount of time to travel to a WPT, complete refueling, and return to homebase. Thus, the QLT did not explicitly take into account the *very important* limitations of actual tanker travel and tanker fuel use. The QLT is incapable of addressing the problem in required levels of detail.

The work of Capehart (2000) extends the capabilities of the QLT through the use of a rudimentary Tabu Search (TS) approach (Glover & Laguna 1997) within the TAP Tool. Due to the fact that only a static short-term memory component with basic move strategies is used in this TS approach, the computational effort required to obtain "good" answers to instances of the problem increases dramatically as the problem size increases. Capehart models the AFRP as an AP with time windows. In addition to this limited modeling perspective, he also makes numerous simplifying assumptions that limit the usefulness of his approach. The most important of these limiting assumptions are (a) a tanker providing fuel to an RG requiring escort must escort the RG to the next WPT, (b) after completing its duties, which may contain at most one refueling, each tanker must return to its home base, (c) light aircraft are escorted over their entire flight, not just when over large bodies of water, (d) the approach does not explicitly account for tanker reuse, but rather corrects time conflicts involving any individual tanker using a penalty structure, and (e) the locations of all WPTs and the amount of fuel required at the WPT are calculated based on a percentage of the full capacity of the RG.

2.2 Literature Associated with Relaxations and Restrictions to the AFRP

Chapter 1 showed how the VRP, JSSP, SCP, and AP could be viewed as special cases of the AFRP. In this section, a brief literature review is presented for each of these classical formulations.

The *VRP*, in its various forms (Barnes & Carlton 1996), is one of

the more widely studied problems contained within the field of combinatorial optimization (Papadimitriou & Steiglitz 1982). This class of problems has been shown to be *NP*-hard (Lenstra & Rinnooy Kan 1981). Christofides (1985) and Golden & Assad (1988) present excellent early reviews of the published work dealing with this class of problems. The first written and formal definition of the VRP occurs in Dantzig & Ramser (1959). The computing power available at the time of that first definition and the eventual demonstration of the VRP as an *NP*-hard problem (Lenstra & Rinnooy Kan 1981), led to the development of a great number of approximation algorithms. Since then, the rapid increase in computing capability has led to the exploration of many variants of the VRP that were previously intractable. The most relevant variant to the AFRP is the multidepot VRP with customer time windows and route length constraints on the vehicles $(MV\overline{H}, MD, VRP, RL, TW)$.

Since it is natural for customers to have preferred delivery times, much work has focused on this area (Lenstra & Rinnooy Kan 1981, Christofides 1985, Baker & Schaffer 1986, Golden & Assad 1988) and more recently (Homberger & Gehring 1999, Chen, Wan & Xu 1998, Chiang & Russell 1997, Gendreau et al. 1997, Kohl & Madsen 1997, Ong, Ang, Goh & Deng 1997, Thangiah, Potvin & Sun 1996, Carlton 1995). The addition of time windows to the basic single vehicle, single depot, VRP yields problem $\gamma = (SV, SD, VRP, --, TW)$. Simply finding a feasible solution to problem γ is *NP*-hard (Savelsbergh 1985, Savelsbergh 1992). Because of this, many proposed solution techniques to this problem maintain feasibility throughout the solution process after an initial feasible solution is found by adding route length restrictions. Adding these restrictions, however, limits a vehicle's operating range. Recent work addressing

this problem includes Rego (1998) and Renaud, Laporte & Boctor (1996).

According to Assad (1988), vehicle routing represents an ideal blend of theory and practice. Unfortunately, despite the continual increases in computing power, practical sized VRP problems are approachable only by using heuristic and metaheuristic techniques. Recent works in that category include an enhanced version of the Generalized Assignment Heuristic (Baker & Sheasby 1999b), an interweaving of local search techniques into a set-partitioning model (Kelly & Xu 1999), and a stochastic programming model with recourse (Savelsbergh & Goetschalckx 1995). Gendreau et al. (1997) provides a more recent overview of works in the field of metaheuristics as applied to the VRP. Further studies using a Genetic Algorithm (GA) include an offshoot of GA known as Evolutionary Strategies (Homberger & Gehring 1999) and the representation of the rolling batch problem as a VRP (Chen et al. 1998). Studies using Greedy Randomized Adaptive Search Procedure (GRASP) include the use of an adaptive strategy (Atkinson 1998) and a general application (Kontoravdis & Bard 1995). The primary metaheuristic found in the literature for the VRP is TS. Recent works implementing the TS framework include the use of a set of mutually exclusive restart neighborhoods (Harder 2000, Kinney 2000), group theory to characterize move neighborhoods (Barnes, Colletti & Neuway 2000b, Colletti & Barnes 2000b, Colletti 1999, Colletti, Barnes & Dokov 1999), heterogeneous vehicles (Gendreau, Laporte, Musaraganyi & Taillard 1999), within a Branch-and-Cut scheme (Augerat, Belenguer, Benavent, Corberan & Naddef 1998), using ejection chains (Rego 1998), with multidepots (Renaud et al. 1996), and with generalized precedence constraints (Nanry & Barnes 2000, Nanry 1998).

Excellent literature reviews of the *Job Shop Scheduling Problem (JSSP)* are contained in Jain & Meeran (1999) and Chambers (1996). The JSSP was considered as early as the 1950's (Jackson 1956, Johnson 1954) and the 1960's saw an emphasis on exact optimization methods. The exact methods reached their zenith in the approaches of Carlier & Pinson (1989), Adams, Balas & Zawack (1988) and Applegate & Cook (1991). During the 1970's and early 1980's, much research was performed in the JSSP complexity arena. By the end of the 1980's, the general understanding of the extreme difficulty of larger instances of the JSSP led to a wider interest in heuristic approaches. This was exemplified by the work of Fisher & Rinnooy Kan (1988) who set forth guidelines for the construction of good heuristic methods. In modern approaches, the JSSP is most often modeled on the disjunctive network formulation of Roy & Sussmann (1964) and Balas (1969). This is true of the

> "Best methods (which) appear to be those encompassing hybrid systems such as local search techniques embedded within a metastrategy that employ a simple neighborhood structure and transcend poor local optimality by allowing non-improving moves" (Jain & Meeran 1999).

The metaheuristic methods first appeared in the simulated annealing approach of Suh (1988) and Matsuo, Suh & Sullivan (1988). Other simulated annealing efforts were reported by Aarts, Van Laarhooven, Lenstra & Ulder (1994), Van Laarhoven, Aarts & Lenstra (1992), Yamaha & Nakano (1996a), and Yamaha & Nakano (1996b). Also reported in the late 1980's and

early 1990's were numerous genetic algorithm approaches such as those offered by Pesch (1993), Della Croce, Tadei & Rolando (1994), and Cheng, Gen & Tsujimura (1996). Other metaheuristic approaches to the JSSP were also reported during this time such as ant optimization (Colorni, Dorigo, Maniezzo & Trubian 1994), GRASP (Resende 1997), and reinsertion algorithms (Werner & Winkler 1995).

The dominant approaches to the JSSP belong to those researchers employing variants of the TS metaheuristic. Among these are the efforts of Taillard (1989), Dell'Amico & Trubian (1993), Barnes & Chambers (1995), and Nowicki & Smutnicki (1996). The technique of Nowicki & Smutnicki (1996) is thought by many to be the most powerful of the methods currently in existence.

As detailed by Balas (1983) and Ceria, Nobili & Sassano (1995), the *Set-Covering Problem (SCP)* is another abundantly studied combinatorial optimization problem with a wealth of practical applications. The SCP is *NP*-hard in the strong sense (Garey & Johnson 1979, Caprara, Fischetti & Toth 1999). Current application arenas generating intense interest are associated with air crew scheduling and railway scheduling where problems can arise that involve thousands of constraints (rows) and millions of decision variables (columns). The best exact methods (all branch-and-bound techniques) exemplified by Balas & Ho (1980), Beasley & Jörnsten (1992), and Balas & Carrera (1996) can solve problems with only several hundred rows and several thousand columns. This lack of capability has led to many efforts in the heuristic arena. Simple greedy heuristics for the SCP have been found to be very

fast but ineffective in solution quality. Balas & Carrera (1996) and Caprara, Fischetti & Toth (2001) conducted an extensive computational comparison of existing heuristic approaches (excluding simple greedy heuristics) to the SCP. Included in this study were the GA approach of Beasley & Chu (1996), the simulated annealing approaches of Jacobs & Brusco (1995) and Brusco, Jacobs & Thompson (1996), and the Lagrangian relaxation with subgradient optimization techniques of Caprara et al. (1999), Ceria et al. (1995), and Balas & Carrera (1996). Caprara et al. (2001) found that the methods using Lagrangian relaxation with subgradient optimization were, as a group, superior and that the technique presented by Caprara et al. (1999) is the overall dominant approach.

The first polynomial method for solving the *Assignment Problem (AP)*, also known as the "Weighted Bipartite Matching Problem", was presented by Kuhn (1955) and Kuhn (1956) when he developed the "Hungarian method." Kuhn's algorithm ran in $O(n^4)$ time. Lawler (1976) reduced Kuhn's method to an $O(n^3)$ algorithm. Derigs (1985) showed that Lawler's approach was equivalent to a successive shortest path algorithm on the network associated with the weighted bipartite matching problem. All current competitive algorithms are based on this network formulation and have $O(n^3)$ time complexity (Dell'Amico & Toth 2000).

The next section provides some of the basic concepts from the realm of abstract algebra that are required in the remainder of this dissertation. Readers already familiar with the fundamentals of group theory and the Symmetric Group on n-letters (S_n) may find it convenient to move directly to the start

of Chapter 3.

2.3 Algebraic Foundations

Group theory, the "algebra of permutations", can powerfully enhance the study and understanding of metaheuristic approaches to combinatorial optimization problems. This statement, illustrated in Colletti (1999), Colletti et al. (1999), Barnes & Colletti (1999), Colletti & Barnes (2000b), and Colletti & Barnes (2000a), is particularly evident when metaheuristic search is applied to Partitioning and Ordering Problems (P|O) like the AFRP. For clarity of explanation, consider the special case of the m-agent Traveling Salesman Problem (TSP) where the agents do not share a common base or depot city (Gilmore, Lawler & Shmoys 1985). Rather, each agent is based at one of the cities in the subtour, or cycle, assigned to that agent, thus yielding the "cyclic" n-city, m-agent traveling salesperson problem (m-CTSP). Let us assume an arbitrary zero diagonal real distance matrix. (When $m = 1$, and the depot is one of the n cities, the m-CTSP becomes the classical 1-TSP.) Some agents may be idle but no more than m cycles are allowed.

Group theory is the foundation of several exact methods of integer programming (Glover 1969, Gomory 1958, Gomory 1965, Gomory 1969, White 1966, Wolsey 1969, Salkin 1975), but it has found limited use in heuristic and metaheuristic methods. In integer programming, the parent group is abelian (commutative). This abelian parent group's many associated factor groups make it easy to construct Gomory cuts. The parent group for heuristics applied to P|O problems, the S_n, is nonabelian with only one small factor group. S_n's

lack of factor groups forces different group-theoretic approaches.

S_n is the group of permutations of the set $\{1, 2, ..., n\}$ of integers from 1 to n (Artin 1991). Specific elements of S_n may be represented in the long form or the cyclic form. The long form appears as a matrix with 2 rows and n columns. The first row represents the index position of a letter within the permutation. The second row represents the letter located at the index position of the first row. The cyclic form appears as a set of cycles delineated by parentheses. Here is the same permutation taken from S_6: first in the long form and then in the cyclic form:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 6 & 5 & 4 & 1 & 2 \end{pmatrix} \longleftrightarrow (1,3,5)(2,6)(4) \equiv (1,3,5)(2,6)$$

Note that "unit" cycles, i.e. (4), are customarily dropped from the cyclic representation.

S_n and its permutations implicitly represent the tours associated with a multidepot TSP where one or more vehicles reside at each depot. Arbitrarily assigning letters 1, 2, and 4 as the depots, then $(1,3,5)(2,6)$ provides the solution depicted in Figure 2.1. That is, one vehicle departs depot 1 and visits customer 3, then customer 5, and then returns to depot 1. Likewise, one vehicle departs depot 2, visits customer 6, and then returns to depot 2. The final vehicle simply remains idle at depot 4.

Group theory (Gaglione 1992, Isaacs 1994) has many practical uses. A *group* is any set and operation which satisfy the properties of *closure*, *associativity*, *identity*, and *inverse* (g^{-1} is the inverse of group element g). The set of

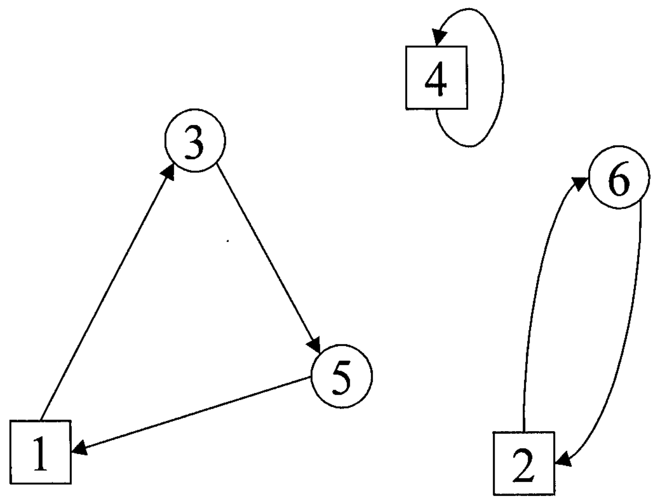

Figure 2.1: Graphical Representation of (1,3,5)(2,6)

all square invertible matrices of order n is a group: multiplying any two such matrices produces another matrix (*closure*); multiplication is *associative*; multiplying a matrix by the *identity* produces the given matrix; and multiplying a matrix by its *inverse* produces the identity. Another familiar group is the integers under addition.

Group theory is applied to the m-CTSP by using S_n, whose elements make up all possible partitions and orderings of the n cities. One group element, or *permutation*, of S_{11} that partitions the 11 cities onto 4 agents is

$$p = (2,3,7)(1,6,5,4)(9,8,11)(10) \equiv (2,3,7)(1,6,5,4)(9,8,11).$$

Permutation p has four subtours or cycles in which each letter is mapped into its successor (denoted $2p = p(2) = 3, 3p = 7, 7p = 2, 10p = 10$). The notation,

$mov(p)$, denotes the letters in the non-unit cycles of p, i.e., for

$$p = (2,3,7)(1,6,5,4)(9,8,11)(10)$$
$$mov(p) = \{1,2,3,4,5,6,7,8,9,11\}.$$

For permutation q, the product pq is the composite function produced by applying p and then q, i.e., for letter x, $xpq = (xp)q$. If $q = (3,7,8,10)(4,9)$ then $3pq = (3p)q = 7q = 8$, and so under pq, 3 is mapped into 8. Permutation multiplication is not commutative since $3qp = (3q)p = 7p = 2$, and, therefore, $pq \neq qp$. Since a permutation represents a 1-1 mapping of its letters $\{1, ..., n\}$ onto itself, multiplication (i.e., function composition) is associative and closed.

The inverse of a permutation reverses each cycle, and the identity permutation maps each letter into itself, i.e., is composed of n unit cycles. Thus, the 4 properties of a group are satisfied and the set of all $n!$ permutations under multiplication is S_n.

Permutations are composed of cycles and an m-cycle has m letters. A 2-cycle is also called a transposition and an n-cycle is a permutation with a single cycle equivalent to a 1-TSP tour. Every permutation is a unique product of disjoint cycles, i.e., cycles sharing no common cities. The number of cycles and the cycle sizes define the permutation's cycle structure and every cycle is a non-unique product of transpositions, i.e.,$(1,2,3,4) = (2,3,4,1) = (1,2)(1,3)(1,4)$ while $(2,3,4,1) = (2,3)(2,4)(2,1)$. If $p^2 = pp$ is the identity, p is an involution.

Other important concepts are:

- If group $H \subseteq$ group G, then H is a subgroup of G, denoted $H \leq G$

- If $H \leq G$ and $g \in G$, then $Hg \equiv \{hg : h \in H\}$ is a right coset of H in G, and gH is a left coset in G
- All left (right) cosets of H exclusively and exhaustively partition G
- If $H \leq G$ and $g \in G$, the Centralizer of g in H is the set of all elements in H that commute with g, i.e., $Cent(H, g) = \{h \in H : h^{-1}gh = g\} = \{h \in H : gh = hg\}$
- A *transversal* on disjoint sets is a collection of elements, made up of precisely one element from each set

A group action G^T uses a group G to partition a set T into mutually exclusive and exhaustive cells called orbits. Regarding group elements as "moves" between elements of T, the elements in the partition of $x \in T$ are those reachable from x by any series of moves. If $g \in G$, then x^g denotes the element in T reached in one step from x via g.

The group action operator is the rule that assigns value to x^g, e.g., conjugation if G and T consist of permutations, the mapping operator if T are letters and G is a permutation group, or similarity products if T and G are $n \times n$ matrices. These examples indicate that elements of G and T may be dissimilar. Finally, a group action must satisfy certain properties in order to be valid (Isaacs 1994), i.e., one cannot freely match any group G with any set T.

Template-Based Neighborhoods

Colletti (1999) and Colletti et al. (1999) describe a group-theoretic process that can transform any permutation in S_n into any other permutation

in S_n. Templates are elements of S_n. Postmultiplying a permutation, p, with the inverse of a splitting template, t, will fragment p. The resulting permutation, $q = pt^{-1}$, will have more cycles than p. Postmultiplying q by a welding template, w, will recombine 2 or more of q's fragments to form another permutation, r, with fewer cycles than q. The cycle structure of r can differ from p. Templates algebraically describe and generalize the classical cross exchange (Taillard, Badeau, Gendreau, Guertin & Potvin 1997) and subtour patching methods (Gilmore et al. 1985, Karp 1979, Bartalos, Dudas & Imreh 1995).

For example, consider the 12-cycle,

$$p = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) \in S_{12}$$

and the splitting template $t = (1, 3, 6, 10, 12)$. $q = pt^{-1}$ yields

$$q = (1, 2)(3, 4, 5)(6, 7, 8, 9)(10, 11).$$

This is the same as removing the arcs {[2,3], [5,6], [9,10], [11,12], [12,1]} from p and allowing the ends of the resulting subpaths to relink to one another. Cycle t is the cycle formed from the p-ordered tails of the subpaths of p that survive the cuts.

In general, if $\{\lambda_i\}$ are the disjoint cyclic factors of p, and if τ_i is a splitting template of λ_i, then

$$q = \Pi\lambda_i\tau_i^{-1} = [\Pi\lambda_i][\Pi\tau_i^{-1}] = p[\Pi\tau_i^{-1}].$$

Thus, if τ is the product of the splitting templates, then $q = p\tau^{-1}$. The above mathematical operations use the fact that disjoint permutations commute: the

τ_i's are disjoint and when $k > i$, τ_i and λ_k are also disjoint. We also know that for any pair of group elements, $x^{-1}y^{-1} = (yx)^{-1}$.

A *welding template*, w, is an m-cycle that unites m disjoint cycles according to the template's letter sequence. Template letters come from distinct cycles, which may include 1-cycles. For example, $\{(1,2,3),(4,5,6,7),(8,9)\}$ are united by $w = (1,4,8)$ to create

$$(1,2,3)(4,5,6,7)(8,9)w = (1,2,3,4,5,6,7,8,9).$$

If $w = (5,2,9)$ then

$$(1,2,3)(4,5,6,7)(8,9)w = (5,6,7,4,2,3,1,9,8).$$

Note that in the resulting cycle, each factor appears as a subpath with the tail specified in the template.

A *joining template* may also be viewed as a product of disjoint welding templates. Using p, λ_i, and τ_i defined earlier, suppose $\{w_k\}$ are disjoint welding templates on the fragmentary cycles. If τ is the product of the τ_i's and if w is the product of the w_i's, then the m-TSP tour q created by fragmenting p and then uniting specified fragments is $q = p\tau^{-1}w$.

As a special case whose derivation involves splitting and welding templates, consider the classical k-Or neighborhood (Carlton & Barnes 1996b). In any m-CTSP tour $p \in S_n$, the k-Or move repositions a k-letter subpath $[t, ..., h]$ after letter x to obtain the new tour q. If $\{t, p(h), p(x)\}$ are distinct and x is not in subpath $[t, ..., h]$, then $q = pr$ where r is the 3-cycle, $r = (t, p(h), p(x))$. The k-Or neighborhood is obtained by computing all such q where x satisfies the stated conditions.

For example, consider

$$p = (1,2,3,4,5,6,7,8)(9,10,11)$$

and its subpath $[3,4,5,6]$. Since

$$r \in R = \{(3,7,1),(3,7,2),(3,7,8),(3,7,9),(3,7,10),(3,7,11)\},$$

the full 4-Or neighborhood obtained by repositioning the subpath throughout p is

$$pR = \left\{ \begin{array}{c} (1,2,7,8,3,4,5,6)\,(9,10,11),(1,3,4,5,6,2,7,8)(9,10,11), \\ (1,2,7,3,4,5,6,8)(9,10,11),(1,2,7,8)(3,4,5,6,9,10,11), \\ (1,2,7,8)(3,4,5,6,10,11,9),(1,2,7,8)(3,4,5,6,11,9,10) \end{array} \right\}$$

This concludes the literature review and a short review of the mathematical background needed for the remainder of this dissertation. The next chapter presents a detailed discussion of the implementation of the adaptive TS methodology that was developed to solve the AFRP.

Chapter 3

Implementation of a Java$^{\text{TM}}$-Based Group Theoretic Tabu Search to the AFRP

This chapter provides a detailed description of a Java$^{\text{TM}}$ language implementation of a GTTS approach to solve the AFRP. The chapter starts by reviewing the basics of object-oriented programming. It then discusses the use of objects within this implementation and the data and methods required for these objects. After the description of each of the AFRP objects, their use with the Symmetric Group solution representation object is provided. Finally, a description of the GTTS is given.

3.1 Basics of Object-Oriented Programming

This section defines Object-Oriented Programming (OOP) and some basic concepts and terms. Readers familiar with OOP may want to skip to section 3.2.

A definition of OOP found at *http://webopedia.internet.com* is repeated here:

> A type of programming in which programmers define not only the data type of a data structure, but also the types of operations

(functions) that can be applied to the data structure. In this way, the data structure becomes an **object** that includes **both** data and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.

One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify.

To perform object-oriented programming, one needs an object-oriented programming language (OOPL). Java, C++ and Smalltalk are three of the more popular languages, and there are also object-oriented versions of Pascal.

Additionally, within *http://webopedia.internet.com* an object is defined as:

> Generally, any item that can be individually selected and manipulated. This can include shapes and pictures that appear on a display screen as well as less tangible software entities. In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data.

Simply put, an object is a set of variables and related methods. The variables contain the data and the methods manipulate the data. Before objects can be used, they must be defined. The definition of what it means to be a particular object occurs within classes.

A class declares the types of information that the object should contain (the variables) and the data manipulations allowed (the methods). A class in this manner represents a *template* from which an object is built.

When a class is used to create an object (that can be selected and manipulated), this process is known as *instantiation.* Instantiation of an object represents the creation of a unique individual object. The term instantiation comes from the idea that every object, at the *instant* it is created, is a unique individual object.

One of the primary advantages to OOP is the reuse of previously defined classes to create subclasses. Subclasses inherit the variables and methods of the source class as well as add their own specific variables and methods. An example of this for the AFRP is shown in Section 3.2.

3.2 AFRP Implementation Objects

The AFRP contains five primary classes (object templates). These are listed below with the types of objects that are associated with them:

- Location: Bases, WPTs
- Aircraft: Light receivers, heavy receivers, tankers

- Receiver Group: RGs requiring escort over water, RGs with no escort required

- Tanker: all of the various types of tanker aircraft, i.e., KC135R, KC10, KC130, and so on.

- Node: Tanker nodes, WPT nodes, Return-to-Base (RTB) Nodes

Figure 3.1 depicts the relationships between the AFRP classes. The Tanker, RG, and Node class templates all contain variables that reference the identification number(s) of objects that are created by the Location class. The RG class template contains a reference to a set of identification number(s) of objects that are created by the Aircraft class. The Node class contains references to the identification numbers associated with its location, the RG associated with this location, and the tanker assigned to service the RG. All tankers are aircraft; therefore, the Tanker class is derived from the Aircraft class. The Tanker class inherits all of the characteristics of a typical aircraft which are joined to additional specific tanker characteristics.

A more detailed discussion of the types of information that each of these class templates require in order to instantiate objects is provided in the next section.

3.2.1 Data Required

The AFRP requires detailed information about:

- the physical locations of bases and WPTs

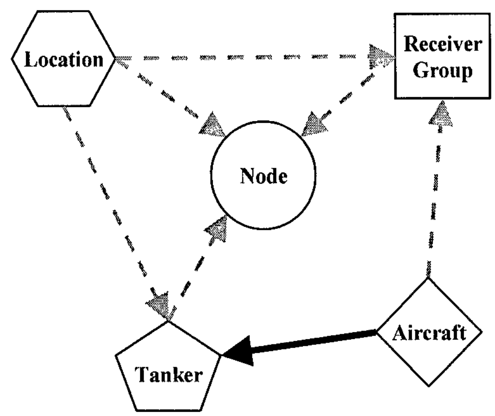

Figure 3.1: Relationships between classes in the AFRP. This figure portrays the relationships between classes in the AFRP. A solid line indicates that the class with the arrow is a direct subclass of the other class (the Tankers class is a subclass of the Aircraft class). A dotted line indicates that the class with the arrow references information from the originating class. For example, each Tanker, RG, and Node objects will contain an identification number referencing a Location object.

- the flight characteristics of aircraft
- the assignment of aircraft to RGs and the stipulation of the flight path that a RG will follow
- the number of tankers available for refueling activities and their associated beddown bases

Each of these sets of data, given in an input source text file, contains the information needed for the specific deployment under consideration. These, in turn, are used to instantiate the necessary Java^{TM}-based AFRP objects to solve the problem.

Locations: Bases, WPTs

Each location object in the AFRP is instantiated using the following information (information not germane to a particular object is left blank):

- Unique Identification Number (ID)
- code name (alphanumeric acronym)
- coordinates (decimal latitude and longitude)
- maximum on ground (MOG)
- whether it is located over open water

After all location objects are instantiated, a *symmetric* distance matrix is generated using the "great circle" distances associated with every pair of locations (Capehart 2000). Figure 3.2 depicts the significant variables and methods of instantiated objects of the Location class.

Aircraft: Light receivers, heavy receivers, tankers

Each aircraft object in the AFRP is instantiated with the following data fields:

- unique ID
- airframe type (light, heavy, tanker)
- nominal true air speed
- total fuel capacity

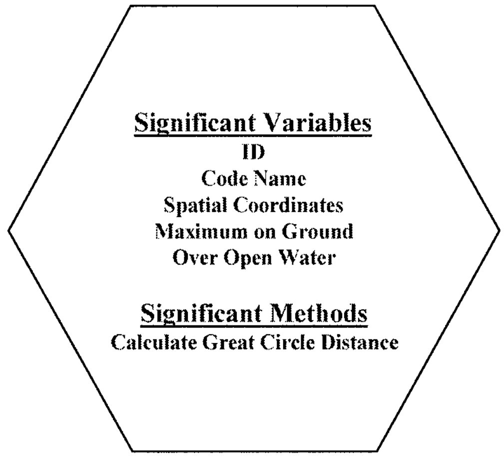

Figure 3.2: Important Location class variables and methods.

- fuel-burn characteristics
  - required fuel reserve
  - nominal fuel used after take-off while climbing to altitude
  - nominal fuel flow coefficient
  - nominal altitude
  - empty weight
  - nominal load weight

This information is also used to complete the instantiation of all RGs and tankers. Figure 3.3 depicts the significant variables and methods of instantiated objects of the Aircraft class.

Tankers

Each tanker object is instantiated as an aircraft object with additional tanker specific information:

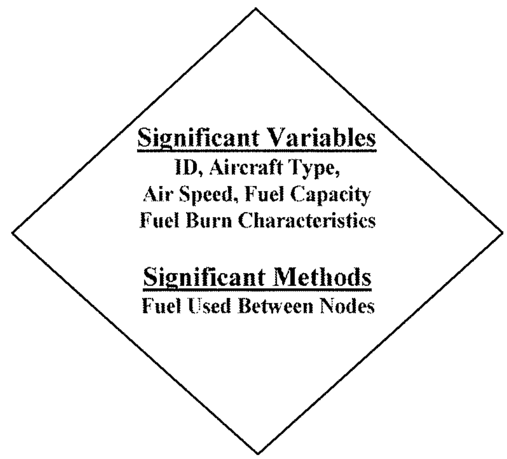

Figure 3.3: Important Aircraft class variables and methods.

- Unique ID (implies tanker and tanker type)
- Off-load capacity
- Tanker solution attributes
  - the WPT nodes served and the times (arrival, service, and departure) associated with this service
  - the amount of fuel burned associated with each WPT pair served and the off-loaded fuel amounts at each WPT served
  - the start and finish times of the tanker for the deployment effort

The tanker solution attribute data fields for a particular tanker remain blank until a solution is found that uses that tanker.

Figure 3.4 depicts the significant variables and methods of instantiated objects of the Tanker class.

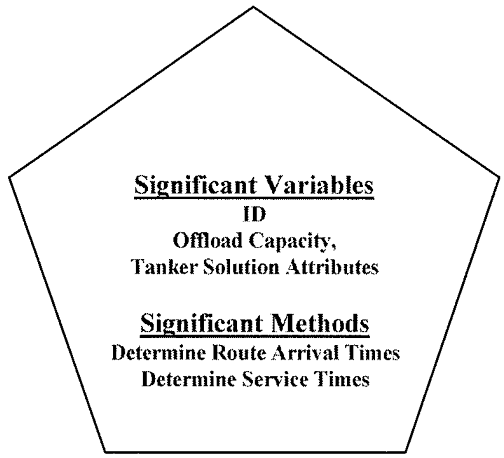

Figure 3.4: Important Tanker class variables and methods.

Receiver Groups: RGs requiring escort over water, RGs with no escort required

Each RG object in the AFRP is instantiated using the following information:

- Unique ID
- Total fuel upload requirement
- List of unique airframe identification numbers defining the aircraft that comprise the RG
- Flight Path Information
  - Starting and ending base location code names or IDs
  - RG flight characteristics
  - Earliest Departure Time (EDT) and Required Delivery Time (RDT) for the RG
- RG solution attributes

Significant Variables
ID, Aircraft Member IDs,
Flight Path Information,
Fuel Upload Requirement,
RG Solution Attributes

Significant Methods
Determine Route Arrival Times

Figure 3.5: Important RG class variables and methods.

- a list of flight path WPT nodes
- the RG's arrival, service, and departure times at each WPT node
- the RG's amount of fuel burned between WPT nodes
- the amount of fuel required to completely refuel each member of the RG.
- the RG's start and finish times

The RG solution attribute data fields for a particular RG remain blank until a solution is found that includes that RG. Figure 3.5 depicts the significant variables and methods of instantiated objects of the RG class.

Nodes: Tanker nodes, WPT nodes, RTB nodes

There are three types of node objects: tanker nodes, WPT nodes, and RTB nodes. The unique indices associated with each node object are used, as explained in Section 3.2.2, to represent and differentiate between the possible

solutions to an AFRP. Node objects may contain the following data:

- a unique ID (which implies the node type)
- the node's spatial location and fuel requirement
- the RG assigned to the node
- precedence relations with a set of other nodes
- whether the node is linked to another node for escort duty
- the tanker assigned to the node
- the fuel demand

For a tanker node, all but the first field are blank. The first field holds the tanker ID. In order to find a feasible solution (where no tanker runs out of fuel), it may be necessary to allow some of the tankers to return one or more times to a base capable of tanker maintenance and refueling. At such a base, the completion of necessary operations, including refueling, allows the tanker to continue servicing WPT's. To allow for this, tanker RTB nodes are instantiated for each active tanker base (as needed). These nodes are characterized by zero fuel demand and no assigned RG or tanker identification number. An RTB node is "employed" to indicate that, at a specific intermediate point in its duty cycle for the current deployment, a tanker must return to either its home base or some other base where maintenance and refueling are performed. Thus an RTB node will possess a unique ID, its spatial location will correspond to

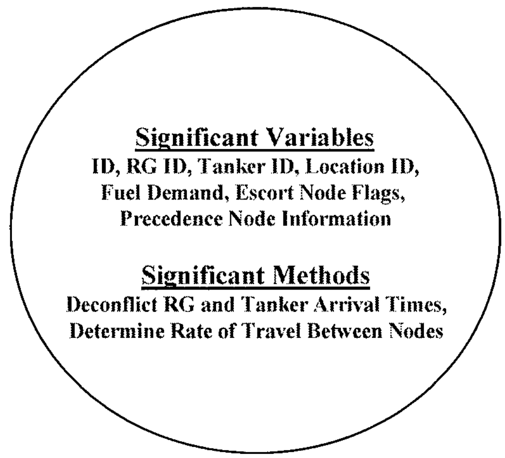

Figure 3.6: Important Node class variables and methods.

the refueling base, and the tanker index of the associated tanker will occupy the sixth field. (The other four fields are blank for an RTB node.)

A WPT node will have the WPT node index, the WPT's latitude and longitude, and the RG's index in the first three fields, respectively. The fourth field will contain the information needed to ensure that the WPT nodes are visited in the correct order along the RG's flight path. If the WPT node is one of a pair both of which are over a large body of water *and* the RG contains one or more light aircraft, the fifth field will indicate that the flight "leg" between the two WPTs requires escort by a tanker. The sixth field will remain blank until a tanker is assigned to the WPT node as part of a generated solution to the AFRP. The seventh field will contain the RG's fuel demand at that WPT.

Figure 3.6 depicts the significant variables and methods of instantiated objects of the Node class.

In solving a particular AFRP, the tanker nodes are generated first and are sequentially assigned indices starting at index 0. The WPT nodes are then created in the ascending order by RG ID and then by ascending order of associated WPT ID's within the RG flight path. For each WPT, its WPT nodes are indexed sequentially starting with the next available index. For every WPT along each RG's flight path, either 1, 2, or 3 WPT nodes are generated.

Only one WPT node is required if both of the following conditions hold true:

- the WPT's fuel demand does not exceed an appropriate predefined limit (currently set at 100,000 lbs)
- the WPT's RG does not require escort to the next WPT

Two WPT nodes are required if either of the following two sets of conditions are true:

1. the WPT's fuel demand exceeds the predefined limit and the WPT's RG does not require escort to the next WPT
2. the WPT's fuel demand does not exceed the predefined limit and the WPT's RG does require escort to the next WPT

If condition 1 holds, the WPT is represented by two WPT nodes where each is assigned one-half of the original WPT's fuel demand. With this WPT representation, a different tanker can be assigned to each WPT node, jointly

satisfying the total WPT demand. (It is currently assumed that no WPT's demand will exceed the combined capability of two tankers.)

If condition 2 holds, the WPT is represented by two WPT nodes where the first node is assigned all of the WPT's demand and the other is assigned a demand of zero. These WPT node creations and assignments are made so that one tanker can perform the refueling function at the WPT and, if appropriate, another tanker can perform the escort duty to the next WPT. (It is assumed that the escorting tanker will provide the refueling function at the next WPT in the RG's flight.)

Three WPT nodes are required if the WPT's fuel demand exceeds the predefined limit and the WPT's RG requires escort to the next WPT. In this case, two of the three WPT nodes serve to allow the required refueling by two different tankers and it is possible that another tanker will assume the escort duty to the next WPT by being assigned to the third WPT node. When the WPT nodes are instantiated, the fuel burned between a RG's adjacent WPTs is calculated and becomes the fuel demand for the latter WPT.

The final type of node generated is the RTB node. RTB are generated as needed and are assigned the next available index.

3.2.2 Instantiation of Tabu Search Objects

Once the AFRP node objects have been created, the tabu search objects are created. In this dissertation, it is assumed that the reader is familiar with classical tabu search techniques as described in Glover & Laguna (1997). Using the OpenTS framework of Harder (2000), the following classes (object templates)

must be defined and created:

- a solution class,
- a move class,
- a neighborhood class,
- an objective function class,
- a tabu list class, and
- an optional aspiration criteria class.

Each of the above classes has an associated parent class obtained from Harder's OpenTS framework that provides the conduit to the tabu search engine. Harder's framework provides the mechanism for implementing the tabu search process as illustrated in Figure 3.7. A complete description of OpenTS is provided at

http://oss.software.ibm.com/developerworks/opensource/coin/OpenTS/

A *solution* object in the AFRP contains a Symmetric Group for Tabu Search (SGforTS) object. Each SGforTS object corresponds to an element *(permutation)* of S_n where n is the cardinality of the complete set of Node objects defined for the current instance of the AFRP to be solved. Each letter in a solution object corresponds to the index of an AFRP Node object. The SGforTS object contains generic methods for creating move neighborhoods for any type of problem while utilizing the methods of the Symmetric Group class.

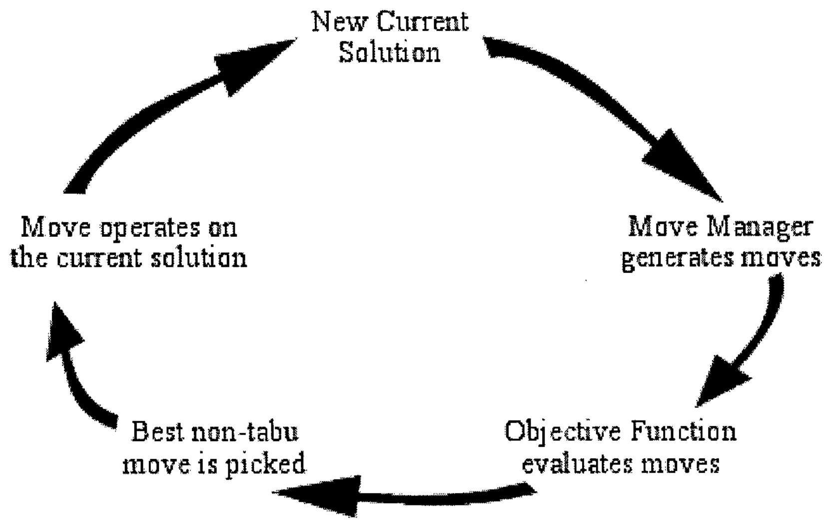

Figure 3.7: An iteration under the OpenTS architecture. This figure shows the typical process flow of a tabu search algorithm. OpenTS provides the framework to implement a tabu search algorithm.

A Symmetric Group object contains the data structures necessary to represent the Symmetric Group using either the long form or cyclic form. The Symmetric Group object also contains the standard group operations of conjugation and multiplication associated with S_n. The relationship of an AFRP solution to the SGforTS and Symmetric Group classes is shown in Figure 3.8.

An **in-depth** description of the Symmetric Group class and the accompanying abstract Group class is presented in Appendices C and D. A java archive file, "jar," with the executable code and the associated Java™ documentation is available from the author or may be downloaded from

http://www.me.utexas.edu/~orie/techrep.html.

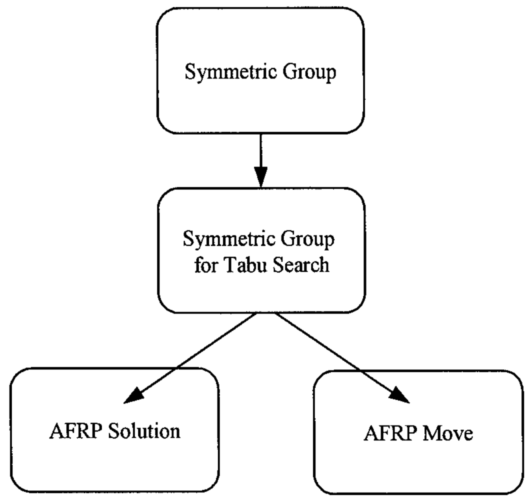

Figure 3.8: Solution and Move Classes. This figure shows the Symmetric Group class as the parent of the SGforTS class. In turn, the solution and move classes contain a SGforTS object as a variable.

The solution object contains the current tanker assignments within the SGforTS object. An example of an individual tanker assignment is given by

"Tanker0 flies to (services) AFRP nodes 15, 16, and 17 and then returns home."

The cyclic representation of tanker 0's assignment would be:

$$(0, 15, 16, 17)$$

with corresponding graphical representation shown in Figure 3.9.

Additional tanker assignments are represented with additional cycles within the SGforTS object. Indeed, all cycles in a solution object will contain a single tanker node which, by convention, is placed in the first position of each cycle in the permutation representation.

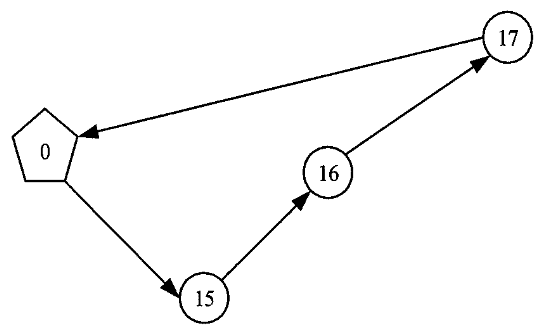

Figure 3.9: An Individual Tanker Route.

The Symmetric Group acts upon itself, so the same class, SGforTS, serves as the template for all AFRP move objects. An AFRP move object, as shown in Figure 3.8 contains a SGforTS object. The SGforTS object represents the move as a permutation that can operate on the AFRP solution permutation to produce a new AFRP solution.

At each iteration, a neighborhood of eligible moves to be evaluated must be passed to the tabu search engine. The neighborhood object in the AFRP contains methods to generate move neighborhoods specific to the AFRP (a detailed description of such move neighborhoods is presented in Section 3.3.2). Once the moves to consider in the current iteration are established, they are passed to the tabu search engine.

The neighborhoods generated by the AFRP are move-based neighborhoods rather than solution-based neighborhoods. A solution-based neighborhood considers solutions reachable by one move. Rather than storing solutions, the moves that transform the current solution to a neighboring solution may be stored. Since function composition in S_n is a bijection of the set $\{1, 2, 3, ..., n\}$ onto itself, the move neighborhoods for the AFRP are composed of Symmetric

Group objects and the move-based neighborhood is a natural extension of this concept.

After receiving a candidate list (Glover & Laguna 1997) of moves to evaluate, the tabu search engine passes each move to the *objective function* object. This object contains the evaluation criteria specific to the AFRP, as described in Section 3.3.3. The objective function object evaluates the move and returns a hierarchical list of values representing the "worth" of implementing a specific move relative to the current solution. The method performs operations necessary to deconflict the schedules of the tankers and RGs that are contained within the objective function object. Once an object has been evaluated, it is passed to the tabu list object for testing.

For each candidate move that exceeds the value of the best move found during a particular iteration, a call to the *tabu list* object is made. As discussed in Section 3.3.4, this call determines whether the implementation of the candidate move upon the current solution is allowed. Once the best available move is selected, that move is implemented and recorded within the tabu structure and the tabu search engine begins a new iteration.

When an aspiration criterion object is instantiated, a move satisfying the associated criterion is accepted regardless of its tabu status. The *aspiration* criterion object allows the user to define a criterion for any particular solution/move combination. The most common aspiration criterion used states that when a new solution is found that is superior to any found earlier in the search, the new solution is accepted regardless of its tabu status.

3.3 Solution Methodology

This section describes the methodology behind the GTTS used to solve the AFRP. The following concepts are described:

- the construction of the initial solution object
- the types of move neighborhoods available for the AFRP
- the *dynamic* neighborhood selection used in the search process
- the objective function criteria used to evaluate a move
- the use of tabu memory.

3.3.1 Instantiation of the Initial Solution

The initial solution object created for the AFRP assigns all WPT nodes to the first tanker (Tanker0) instantiated by the Java program. Usually, this will produce a highly infeasible solution (i.e., the capacity of the tanker will be exceeded). If a problem had 15 tankers and 32 WPT nodes, the initial solution would be represented by:

$$(0, 15, 16, 17, \dots, 44, 45, 46) \tag{3.1}$$

To overcome the infeasibility of the initial solution, an initial set of moves using a Tanker Insert Move Neighborhood (TKI) is generated using the remaining available tankers and inserting them within the current employed tanker's WPT node assignments. The insertion point is strongly influenced by the requirement that some RGs must be escorted over open water. An example

of a tanker insert move, given the initial solution (3.1), is "Insert Tanker5 in front of Node25". Using permutation multiplication, this move is represented by $(0, 5, 25)$.

Performing the multiplication splits the Tanker0 assignment as follows (Tanker0 is assigned nodes 15 through 24 and Tanker1 is assigned nodes 25 through 46):

$$\begin{aligned}&(0, 15, 16, \ldots, 24, 25, 26, \ldots, 46) \otimes (0, 5, 25)\\&= (0, 15, 16, \ldots, 24)(5, 25, 26, \ldots, 46)\end{aligned}$$

Placement of the remaining tankers continues until there are no available tankers or until a feasible solution is obtained. For example, if there were 14 required escort arcs between the 32 WPTs, the 15 tankers might be assigned to the WPTs as follows:

$$\begin{aligned}&(0,15)(1,16,17)(2,18,19)(3,20,21)(4,22,23,24)(5,25,26)\\&(6,27,28)(7,43,44)(8,29,30)(9,45,46)(10,31,32,33,34)\\&(11,35,36)(12,37,38)(13,39,40,41,42)\end{aligned} \tag{3.2}$$

In this solution, 14 tankers are used before the Tanker Insert Move Neighborhood (TKI) is no longer used.

The choice of starting with a single tanker and constructing a solution by inserting tankers provides a number of benefits. The first benefit is that the initial solution is very easy to construct. The second benefit is the use of the AFRP objectives described in Section 3.3.3 to direct the insertion of tankers as opposed to an arbitary construction heuristic. The use of an

arbitrary construction heuristic may not account for the AFRP specific objectives. By allowing the GTTS to determine the placement of the tankers base on the AFRP objectives, the GTTS ensures that beneficial assignments are maintained at each iteration.

3.3.2 Dynamic Neighborhood Selection

Once the initial TKI has performed its function, additional move neighborhoods are invoked based on the current search status and solution. The move neighborhoods were determined by a combination of past successes for VRPs (Carlton 1995) and necessity during the development of this research. These neighborhoods include:

1. a Return To Base Insert Move Neighborhood (RTBI),
2. a Restricted Insert Move Neighborhood (RI),
3. an Escort Pair Insert Move Neighborhood (EPI),
4. a Return To Base Delete Move Neighborhood (RTBD),
5. a Tanker Swap Move Neighborhood (TKS),
6. a Restricted Swap Move Neighborhood (RS), and
7. a Return To Base Swap Move Neighborhood (RTBS).

Each move neighborhood creates an array of eligible *move* objects to be evaluated by the *objective function* object. All move neighborhoods use

permutation multiplication, which allows the search to investigate AFRP solutions from the different appropriate conjugacy classes in S_n. In the following sections, each move neighborhood cited above will be described with the conditions that cause them to be invoked.

The Return To Base Insert Move Neighborhood (RTBI)

After the TKI strategy is complete, the solution may still be infeasible, primarily due to a lack of tanker capacity. To combat this infeasibility, the RTBI neighborhood is used. Choosing a RTBI move reduces infeasible fuel shortage at the cost of delaying one or more RGs, i.e., the tanker is unavailable until it has been refueled and allowed to reenter service.

The RTBI neighborhood is implemented by creating an RTB node for each active tanker base in the deployment scenario which can perform refueling and other required operations on tankers. These RTB nodes may be inserted within each of the current tankers' assignments. Consider this example of an RTB node being placed in a tanker's assignment for solution (3.2): "Insert a return to base node for Tanker4 before Node24." The move that will accomplish this is $(24, 49)$.

The only affected cycle in (3.2) contains Tanker4, so only changes in *that* cycle need to be shown. (This practice of showing only the affected cycles will be continued throughout the rest of this chapter.) As illustrated in Figure 3.10, performing the cited move yields

$$(4, 22, 23, 24) \otimes (24, 49) = (4, 22, 23, 49, 24) \tag{3.3}$$

After the initial solution is achieved, the RTBI neighborhood will be

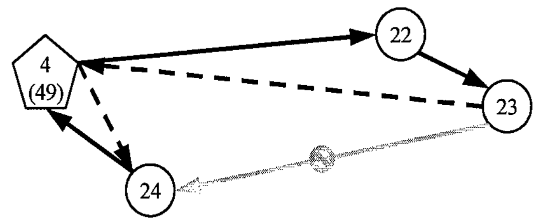

Figure 3.10: Return to Base Insert Move. This figure shows how Tanker4's original route (shown in solid lines) is altered when it is allowed to return to base before continuing service. The dotted lines indicate the new arcs that replace the shaded arc marked with $\oslash$.

invoked whenever the current solution is infeasible. See Section 3.3.3 for a more detailed description of an infeasible solution to the AFRP.

The Restricted Insert Move Neighborhood (RI)

Following initial solution construction, the RI neighborhood is invoked at every iteration of the search and allows an individual letter to be inserted in a different position in its cycle or to be inserted in another tanker's cycle. An RI move can either reorder a tanker's assignment or change the partitioning of the letters among the tankers. The "restriction" on this move neighborhood limits the allowable "distance" that a letter can be moved within the current permutation solution representation. "Distance" is defined as the number of positions a letter may move from its current position either to the left or right. For the results presented in this dissertation, the distance is set at 5 positions. This parameter setting and similar parameter settings discussed below were found through empirical experimentation for the example problems of Chapter 4.

Consider the following two examples of moves with this neighborhood.

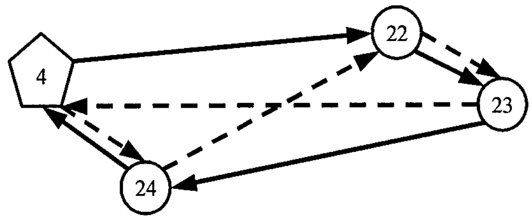

Figure 3.11: Restricted Insert Move With One Tanker. This figure depicts the change in Tanker4's original route (solid lines) when Node24 is moved in front of Node22. The new route is indicated by the dashed line.

The first involves inserting a letter within its current cycle. Using solution (3.2), we "Insert Node24 in front of Node22 within Tanker4's current assignment." As illustrated in Figure 3.11, the move and changes in Tanker4's assignment are given by

$$(4, 22, 23, 24) \otimes (4, 22, 24) = (4, 24, 22, 23)$$

Using solution (3.2), the second example inserts a letter into another tanker's cycle., i.e., "Insert Node24 from Tanker4's assignment in front of Node25 in Tanker5's assignment".

As shown in Figure 3.12, the move and changes in the tankers assignments are:

$$(4, 22, 23, 24)(5, 25, 26) \otimes (4, 25, 24) = (4, 22, 23)(5, 24, 25, 26)$$

The Escort Pair Insert Move Neighborhood (EPI)

After the initial solution construction, the EPI neighborhood is invoked for every iteration of the search. EPI is similar to the RI. The difference consists

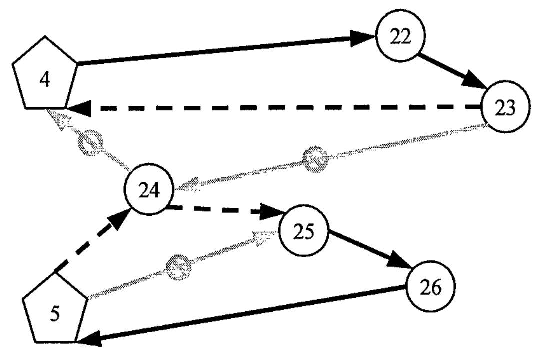

Figure 3.12: Restricted Insert Move Between Tankers. This figure shows how the original route assignments (solid lines) for tankers 4 and 5 change when Node24 is removed from Tanker4's route and placed in Tanker5's. The dotted lines indicate the new arcs flown by the tankers and the shaded arcs marked with ⊘ are no longer used.

of the EPI neighborhood inserting two letters associated with an escort arc instead of a single letter.

Consider the following two clarifying examples: first, for solution (3.2), we insert a pair within their current cycle, i.e., "Insert the pair (31,32) after Node34 in Tanker10's assignment."

As Figure 3.13 depicts, the move and changes in Tanker4's assignment are

$$(10, 31, 32, 33, 34) \otimes (10, 31, 33) = (10, 33, 34, 31, 32)$$

The second example, presented in Figure 3.14, insert the pair into another tanker cycle. For solution (3.2), "Insert the pair (31,32) after Node46 in Tanker9's assignment."

The move and changes in the tankers' assignments are:

$$(9, 45, 46)(10, 31, 32, 33, 34) \otimes (9, 31, 33) = (9, 45, 46, 31, 32)(10, 33, 34)$$

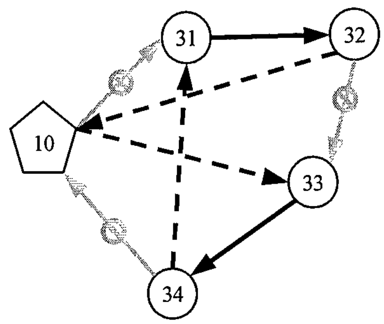

Figure 3.13: Escort Pair Insert Move With One Tanker. This figure represents how Tanker10's original route (solid lines) is changed by an escort pair insert move. The dotted lines indicate the new arcs created when nodes 31 and 32 are positioned after Node34. The shaded arcs marked with ⊘ are no longer used.

Both the RI and the EPI neighborhoods can be viewed as variants of the traditional k-Or move where the subpath consists of the pair of letters for the escort pair.

The Return To Base Delete Move Neighborhood (RTBD)

RTBI moves help the search progress towards feasibility while increasing the number of "letters" being used in the solution. Once the number of letters reaches 1.5 times the original number of letters, the RTBD move neighborhood is invoked. This move neighborhood removes any extra, i.e., nonbeneficial, RTB nodes from the solution.

An example of this type of move is illustrated in Figure 3.15. In solution (3.3), "Remove the Return To Base Node49 from Tanker4's assignment."

The move and changes in Tanker4's assignment are:

$$(4, 22, 23, 49, 24) \otimes (24, 49) = (4, 22, 23, 24)$$

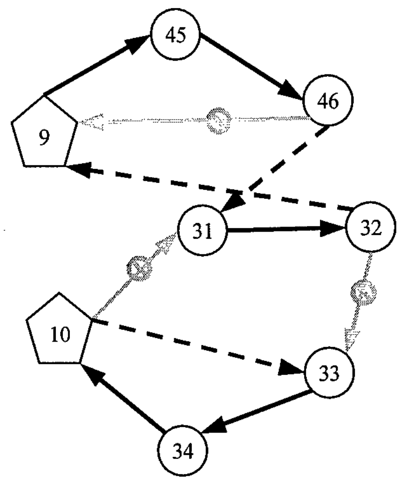

Figure 3.14: Escort Pair Insert Move Between Tankers. This figure represents how the original routes (solid lines) of tankers 9 and 10 are changed by an escort pair insert move. The dotted lines indicate the new arcs created when nodes 31 and 32 are positioned after Node46. The shaded arcs marked with $\oslash$ are no longer used.

The Tanker Swap Move Neighborhood (TKS)

In addition to the RTBD neighborhood, the TKS neighborhood is invoked when the number of nodes in the solution has grown to 1.5 times the original number of letters. This neighborhood allows idle tankers to be exchanged with active tankers (from different bases) to lessen travel and fuel usage.

An example of a TKS move, pictured in Figure 3.16, presumes we are given solution (3.2) and "swap Tanker14 for Tanker10".

The move and changes in the solution are:

$$(10, 31, 32, 33, 34) \otimes (10, 14, 31) = (14, 31, 32, 33, 34)$$

Additionally, the TKS neighborhood also allows an idle tanker to be exchanged with an active tanker that has been relocated to the idle tanker's

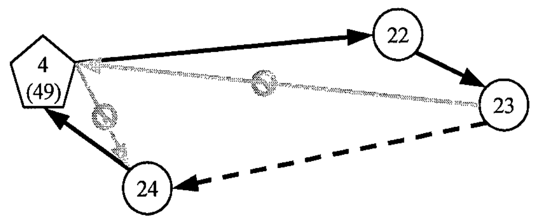

Figure 3.15: Return To Base Delete Move. This figure shows how Tanker4's original route (solid lines) changes when the the return to base node (Node49) is removed. The dotted line indicates the new arc of the route and the shaded arcs marked with ⊘ are no longer used.

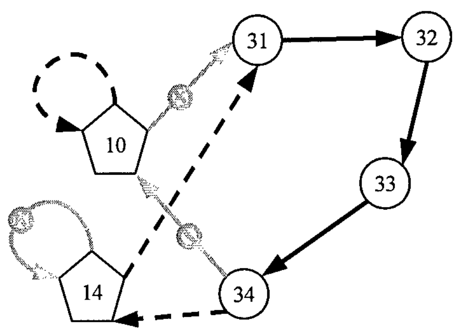

Figure 3.16: Tanker Swap Move. This figure shows how the original routes (solid lines) of tankers 10 and 14 are changed by swapping the two tankers. The dotted lines represent the new arcs added and the shaded arcs marked with ⊘ are no longer used.

beddown base. This may occur in two ways. First, a tanker may relocate, provide service and then return to an active tanker base (which differs from the relocation base and beddown bases). Second, the tanker relocates, provides service, and then returns to the relocation base.

An example of the first case, shown in Figure 3.17, is "Swap idle Tanker10 for active Tanker7." Tanker7 has relocated to Tanker10's beddown base prior to servicing WPTs. After servicing nodes 36 and 37, Tanker7 relocates to another active tanker base. Suppose the original assignment for

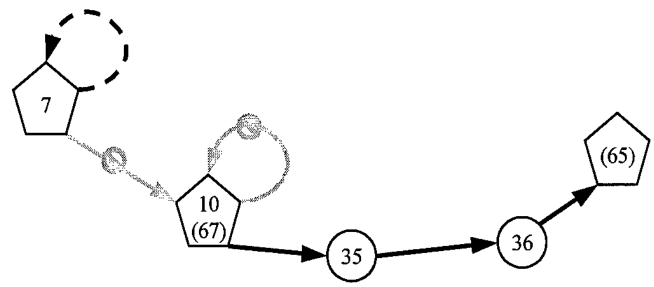

Figure 3.17: Tanker Swap Move With One Return To Base Node. This figure shows how the original routes (solid lines) of tankers 7 and 10 are changed when tanker 10 is swapped for Tanker7 and RTB Node67. The dashed lines indicate the new arcs created and the shaded arcs marked with $\oslash$ are no longer used.

Tanker7 is:

$$(7, 67, 35, 36, 65)$$

The move and change is:

$$(7, 67, 35, 36, 65) \otimes (7, 10, 35, 67) = (10, 35, 36, 65)$$

Note that an RTB node has also been removed from activity.

Figure 3.18 is an example of the second case, "Swap idle Tanker5 for active Tanker13." Tanker13 has relocated to Tanker5's beddown base, provided service, and then returned to Tanker5's beddown base. Suppose the original assignment for Tanker13 is:

$$(13, 51, 41, 42, 57)$$

The move and changes are:

$$(13, 51, 41, 42, 57) \otimes (5, 41, 51, 13, 57) = (5, 41, 42)$$

This time two RTB nodes have been removed.

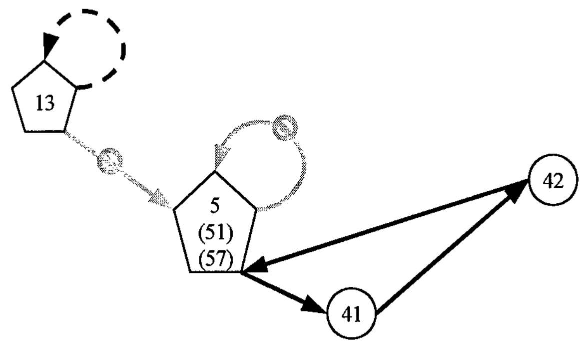

Figure 3.18: Tanker Swap Move With Two Return To Base Nodes. This figure shows how the original routes (solid lines) of tankers 5 and 13 are changed when Tanker5 is swapped for Tanker13 and RTB nodes 51 and 57. The dashed lines indicate the new arcs created and the shaded arcs marked with ⊘ are no longer used.

The Restricted Swap Move Neighborhood (RS)

During the search, if a specified number of iterations have passed (20 iterations for this research) without a new best solution being identified, the RS neighborhood is invoked. This move neighborhood allows an individual letter within a tanker's assignment to be swapped either with a letter in its cycle or with a letter in another tanker's cycle. This has the effect of maintaining the current cardinality of the partitioning of the letters amongst the tankers. The "restriction" of this move neighborhood limits the allowable distance (5 positions for this research) that any letter can be moved (see section 3.3.2).

Consider the following two examples of RS moves for solution (3.2). First, swap a letter with another in its current cycle, "Swap node 31 for node 34 in Tanker10's assignment."

The move and changes of Tanker10's assignment are:

$$(10, 31, 32, 33, 34) \otimes (10, 32)(31, 34) = (10, 34, 32, 33, 31)$$

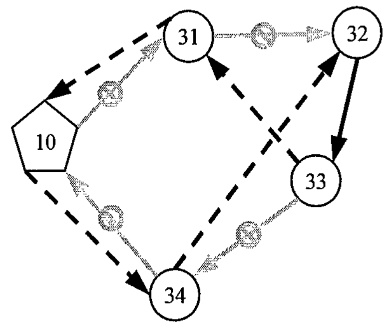

Figure 3.19: Restricted Swap Move With One Tanker. This figure shows how Tanker10's original route (solid lines) is changed by the swapping of positions by nodes 31 and 34. The dashed lines indicate the new arcs created and the shaded arcs marked with $\oslash$ are no longer used.

Figure 3.19 depicts the changes in Tanker10's assignment.

Second, we swap a tanker's assigned letter with another tanker's assigned letter, "Swap Node34 from Tanker10's assignment with Node24 of Tanker4's assignment."

The resultant changes in solution (3.2) are:

$$(4, 22, 23, 24)(10, 31, 32, 33, 34) \otimes (4, 10)(24, 34)$$
$$= (4, 22, 23, 34)(10, 31, 32, 33, 24)$$

Figure 3.20 depicts the old and new route assignments for tankers 4 and 10.

The Return To Base Swap Move Neighborhood (RTBS)

In addition to the RS neighborhood, the RTBS neighborhood is also invoked when a specified number of iterations (20 for this research) have passed without a new best solution being identified. This move neighborhood allows the

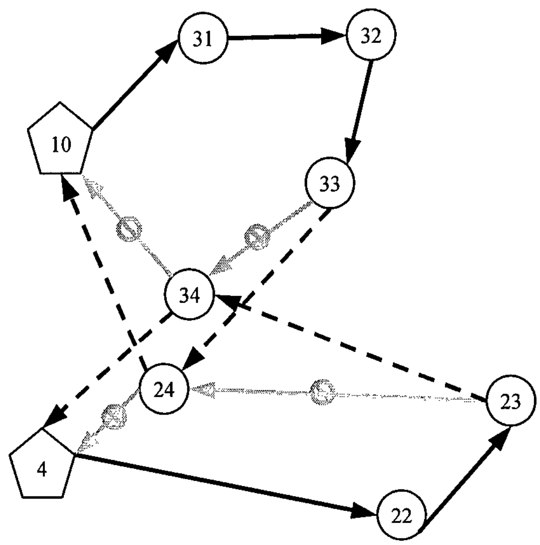

Figure 3.20: Restricted Swap Move Between Tankers. This figure shows how the original routes (solid lines) of tankers 4 and 10 are changed when nodes 24 and 34 are exchanged. The dashed lines indicate the new arcs that are created and the shaded arcs marked with $\oslash$ are no longer used.

RTB nodes to be exchanged with other RTB nodes. This allows the solution to adjust the locations of the RTB nodes to fit the current set of tanker assignments.

An example of an RTBS move is "Swap node 48 for node 56 in Tanker13's assignment." If Tanker13's assignment was $(13, 34, 48)$, the move and changes in Tanker13's assignment are:

$$(13, 34, 48) \otimes (13, 48, 56) = (13, 34, 56)$$

Figure 3.21 depicts Tanker13's old and new route assignments.

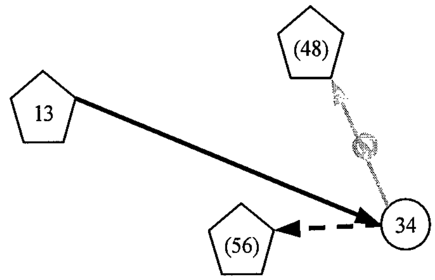

Figure 3.21: Return To Base Swap Move. This figure shows how Tanker13's original route (solid lines) changes when the return to base nodes 48 and 56 are swapped. The dashed lines indicate the new arcs created and the shaded arcs marked with ⊘ are no longer used.

3.3.3 Move Evaluations

Move evaluations for the AFRP are concerned with whether or not a move yields a feasible solution and with whether or not a move yields a superior solution. The AFRP objective function, as previously noted, is multicriteria and hierarchical. For the AFRP, creating an assignment and schedule of tankers to RGs that can actually be flown without the tankers or the aircraft within the RGs running out of fuel is significantly more important than the relative value conditions associated with tanker and fuel use.

Feasibility and "Bad" Choice Conditions

There are a number of feasibility conditions for the AFRP. Each condition can be either physical (i.e., tanker capacity, unfulfilled RG demand), organizational (i.e. fighter aircraft must be escorted while over open water), or temporal (i.e. RG refuelings must maintain a precedence relationship). In addition, certain conditions represent logical misconstructs that imply inferior (non-optimal) solutions (i.e., a tanker being assigned to fly to another base and then returning

to its home base first without serving an RG). These conditions include the following (in hierarchical order of importance):

1. the number of escort arcs not assigned to a tanker (organizational),
2. the number of RG demand nodes not assigned to a tanker (physical),
3. the number of RG demand nodes serviced out of precedence order (temporal),
4. the number of "bad" tanker assignments (logical), and
5. the amount of required fuel not supplied (physical).

   Any occurrences of the above conditions are noted and counted, one at a time, during the evaluation of an AFRP solution.

Unescorted RGs If an RG contains any light aircraft, it must be escorted by a tanker while flying over large bodies of water. If light aircraft are present, all adjacent WPT pairs where *both* WPTs are over open water define an arc (leg) of the RG flight path that must be escorted, i.e., an escort arc. Within an AFRP solution, each identified WPT escort "pairing" must be represented as adjacent nodes within a tanker assignment. The number of unescorted arcs is returned for the purpose of objective function evaluation.

Uncovered RG Demand For every RG, a set of WPTs with fuel demands are generated. Within the current tankers' assignments, feasibility requires that the entire set of WPT fuel demands be covered. During the TS process,

it is possible to have solutions that leave some WPT fuel demands uncovered. The number of uncovered WPT fuel demands is returned.

Misordered Tanker Services For every tanker assignment, the order of service for the assigned WPT fuel demands is extremely important. A tanker can not first provide service to a RG at a WPT and then service another WPT along the RG's flight path that has already been passed. The number of misordered WPT fuel demands is returned.

Bad Tanker Assignments As the tabu search progresses, there are a number of conditions that logically should not occur. These conditions include:

a. "Tanker base hopping" (adjacent RTB nodes)

b. 2 tankers within a single cycle in the AFRP solution object, and

c. a tanker providing repeat servicing to the same RG.

Tanker base hopping occurs during the search when return to base nodes become adjacent to one another within a tanker's assignment. This has the effect of repositioning the tanker from one location to another without providing any useful service. The number of adjacent return to base nodes is recorded.

During the search, a tanker may be assigned within another tanker's route. For the purposes of this model, a tanker must either be providing service or receiving service as part of a RG, but not both. The number of tankers assigned within another tanker's route is recorded.

A tanker providing repeat servicing captures the condition that a tanker could be assigned to provide service to a specific RG, return to an active tanker base, and then provide service to the same RG. This pattern is considered undesirable since a tanker, when it returns to an active tanker base, must remain at that base for a minimum amount of service time (4 hours for this research). During this service, the RG is continuing along its flight path and, more than likely, can not be feasibly serviced again by the same tanker. The number of tanker repeat servicings is recorded.

Infeasible Fuel Usage The amount of fuel required by any tanker assignment includes the amount of fuel burned by the tanker during flight and the amount of fuel off-loaded to RGs at WPTs. If the amount of fuel exceeds the fuel usage capacity of the tanker, then an amount of infeasible fuel use equal to the difference between the used fuel and the available fuel is returned. Additionally, any uncovered WPT fuel demands are also included as infeasible fuel usage. The total amount of infeasible fuel used is returned.

Value Conditions

There are a number of value conditions that are considered. Each condition can be classified as temporal (i.e., a RG arriving at its destination base past the desired arrival time, the amount of time spent in "orbit", and tanker flight time) or physical (i.e., tankers used, tanker distance traveled, and tanker fuel consumed). *Value* conditions include the following (in hierarchical order of importance):

6. the amount of delay time (temporal),

7. the amount of RG late arrival time (temporal),
8. the number of tankers exceeding the allowed maximum on ground (MOG) at an active tanker base (physical),
9. the number of tankers used (physical),
10. the amount of tanker mission time (temporal),
11. the amount of tanker distance flown (physical),
12. the amount of fuel used by tankers (physical),
13. the amount of fuel off-loaded by tankers (physical), and
14. the amount of fuel used by the RGs (physical).

"Orbit" times During the search, scheduled times for a tanker's arrival to an assigned WPT may not coincide with the RG's arrival at that WPT. When this occurs, either the tanker must "orbit" (fly in circles) until the RG arrives or vice versa. In either case, the total amount of time spent in "orbit" by all tankers and RGs is returned.

RG late arrival It is not always possible for all RGs to reach their final destinations by their desired arrival times. The amount of time past the requested arrival time is recorded for each RG. The sum of all RG late arrivals is returned.

MOG restriction Every base has a limited amount of space to park aircraft. This condition notes whether this capacity is ever exceeded during the deployment. To check this condition, an event list for each active tanker base storing arrivals and departures is compiled. The MOG condition for each base is checked at the occurrence of any event in its event list. The number of tankers exceeding the allowable MOG is returned.

Tankers used The actual number of tankers assigned to RG WPTs is returned.

Tanker mission time For each tanker, the difference between its start time and end time is determined. The sum of all the differences in the current solution is returned.

Tanker distance flown For each tanker, the total distance traveled between locations is computed. The sum of all the distances in the current solution is returned.

Tanker fuel usage For each tanker, the amount of fuel used between locations is computed. The sum of all the fuel used in the current solution is returned.

Tanker fuel delivered For each tanker, the amount of fuel delivered to RGs at WPTs is computed. The sum of the delivered fuel in the current solution is returned.

RG fuel usage For each RG, the amount of fuel used between locations is computed. The sum of all the fuel used in the current solution is returned.

3.3.4 Tabu Structure

Part of the strength of tabu search comes from its ability to retain a "memory" of what solutions have been previously visited or what moves have been previously selected. The memory structure used in this tabu search stores the moves that have been previously selected and changes the length of the "short term memory" (tabu tenure) using an adaptive procedure.

Memory structure

Moves are stored in a square matrix of dimension equal to the number of solution nodes present. As moves are selected, the letters moved are recorded in the matrix. For each letter moved, the row index is the letter moved, the column index is the image of the letter moved, and the value placed in the matrix is the current iteration plus the current tabu tenure. The following is an example of a move placed in the tabu matrix. Assuming $n = 3$, initially

$$tabuMatrix = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Given an initial $tabuTenure = 7$ and $move = (1, 2, 3)$ with the current iteration $= 0$

$$tabuMatrix = \begin{bmatrix} 0 & 7 & 0 \\ 0 & 0 & 7 \\ 7 & 0 & 0 \end{bmatrix}$$

When return to base nodes increase the number of solution nodes, the tabu matrix is redimensioned to include the new letters and the current values

within the matrix are retained.

Adaptive Procedure for Changing Tabu Tenure

As the search progresses, the tabu tenure is adaptively modified based on the status of the current solution (Chambers & Barnes 1996, Dell'Amico & Trubian 1993). If the current solution is the best solution found so far, the tabu tenure is reset to a specified default value (7 for this research). If the current solution is a better move than the last, but not the best solution so far, the tabu tenure remains at its current value. If the current solution is not better than the previous move, the tabu tenure is increased by one.

3.3.5 An iteration in the GTTS

The previous sections described the concepts that defined the GTTS for the AFRP. This section discusses the activities and computations that take place during an iteration of the GTTS. This discussion will follow the steps depicted in Figure 3.7.

Move Generation

The GTTS begins with a current solution. Based on the characteristics of this solution and the progress of the tabu search, move-based neighborhoods are generated. The sizes of the neighborhoods will vary as the search progresses and are discussed in more detail in the next sections.

Tanker Inserts For the generation of Tanker Insert moves, the tankers are placed in a "pool" for each beddown base. If a beddown base has unassigned

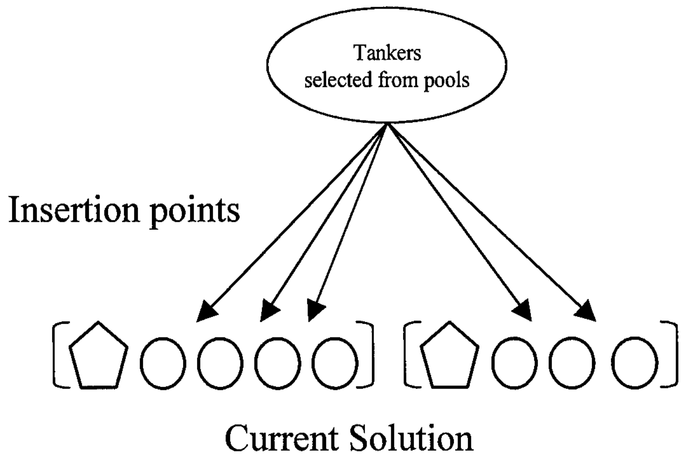

Figure 3.22: Tanker Insert Move Generation. This figure illustrates the insertion points for unassigned tankers within a current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

tankers in its pool, then one of the unassigned tankers is selected. For each selected tanker, moves that insert the selected tanker into the current solution are generated. Available insertion points start at the second waypoint node assigned to a tanker within the current solution and continue up to the last waypoint node as depicted in Figure 3.22. Placing tankers at the first waypoint and after the last waypoint has the effect of swapping out the current tanker for an unassigned tanker and is disallowed in this neighborhood. The total number of tanker insert moves generated equals the number of unassigned tanker pools selected times the number of insertion points where the number of insertion points = the number of waypoint nodes - number of tankers in current solution.

Return-to-Base Inserts The generation of RTB Insert Moves is very similar to the generation of Tanker Insert moves. For each beddown base, a RTB

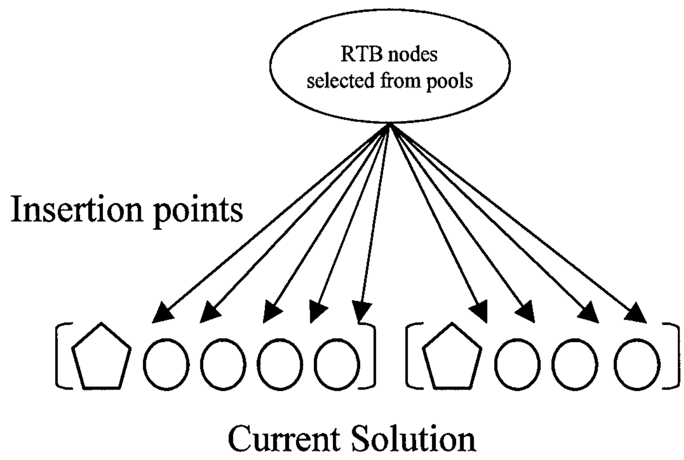

Figure 3.23: RTB Insert Move Generation. This figure illustrates the insertion points for unassigned RTB nodes within a current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

node is selected from a pool of unused RTB nodes. If a beddown base does not have any unused RTB nodes, a new RTB node is created. For each selected RTB node, moves that insert the RTB node into the current solution are generated. Available insertion points start at the first waypoint node assigned to a tanker within the current solution and continue until after the last waypoint node. Figure 3.23 provides an illustration of the insertion points. The total number of RTB inserts generated equals the number of beddown bases times the number of insertion points where the number of insertion points = the number of waypoint nodes + number of tankers in the current solution.

Restricted Inserts For each waypoint node in the current solution, a restricted set of insert moves is generated. The depth of this restriction for this research is 5 positions to the left or right of each waypoint node as depicted in Figure 3.24. The total number of moves generated equals the number of

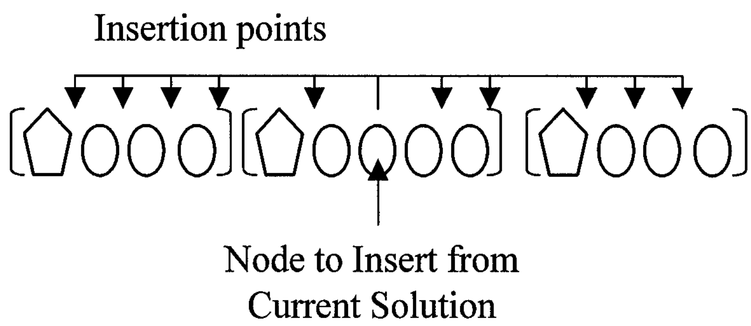

Figure 3.24: Restricted Insert Move Generation. This figure illustrates the insertion points for a waypoint node in the current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

waypoint nodes times twice the depth.

Escort Pair Inserts For each escort pair in the current solution, a restricted set of escort pair insert moves is generated. The depth of this restriction for this research is 5 positions to the left or right of the selected waypoint node as shown in Figure 3.25. The total number of moves generated equals the number of waypoint node escort pairs times twice the depth.

Return-to-Base Deletes For each RTB node in the current solution, a move that removes the RTB from the current solution is generated. The total number of moves generated equals the number of RTB nodes in the current solution.

Tanker Swaps For each beddown base that has unassigned tankers in its tanker pool, a move that exchanges an assigned tanker from a different beddown base within the current solution is generated as illustrated in Figure

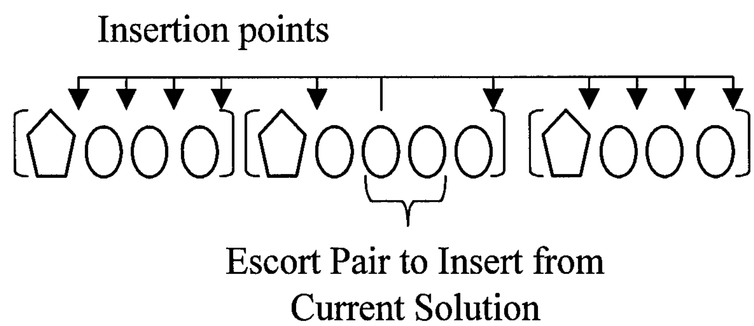

Figure 3.25: Escort Pair Insert Move Generation. This figure illustrates the insertion points for an escort pair in the current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

3.26. The total number of tanker swaps generated equals the number of beddown bases with unassigned tankers times the number of tankers in the current solution.

Restricted Swaps For each waypoint node in the current solution, a restricted set of swap moves is generated. The depth of this restriction for this research is 5 positions to the left or right of each waypoint node as depicted in Figure 3.27. The total number of swaps moves generated equals the number of waypoint nodes times twice the depth.

Return-to-Base Swaps For each beddown base, a RTB node is selected from a pool of unused RTB nodes. If a beddown base does not have any unused RTB nodes, a new RTB node is created. For each selected RTB node, a move that exchanges an RTB node from a different beddown base within the current solution is generated as shown in Figure 3.28. The total number of moves generated equals the number of beddown bases times the number of

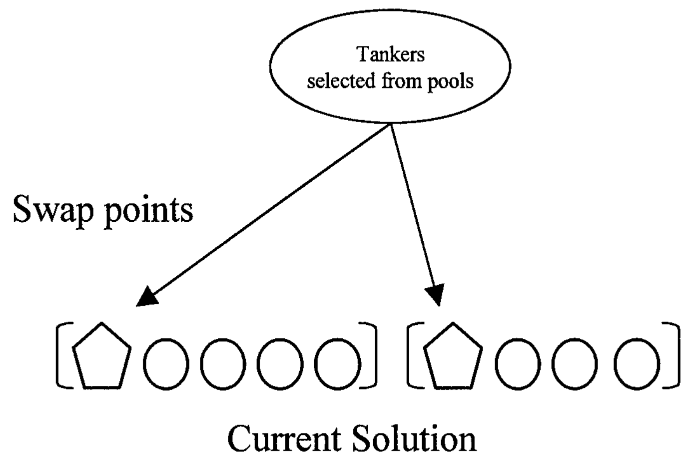

Figure 3.26: Tanker Swap Move Generation. This figure illustrates the swap points for an unassigned tanker within the current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

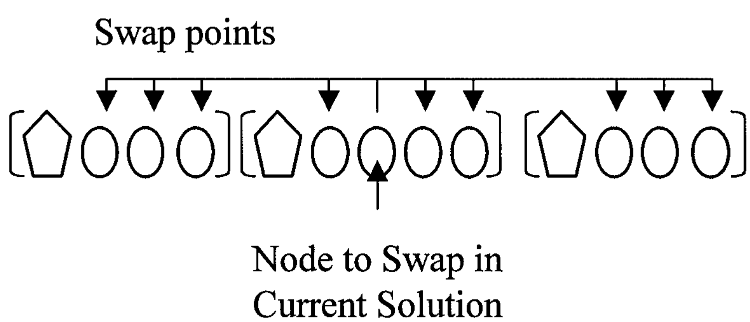

Figure 3.27: Restricted Swap Move Generation. This figure illustration the swap points for a waypoint node in the current solution. Assigned tankers are represented by pentagons and waypoint nodes by circles.

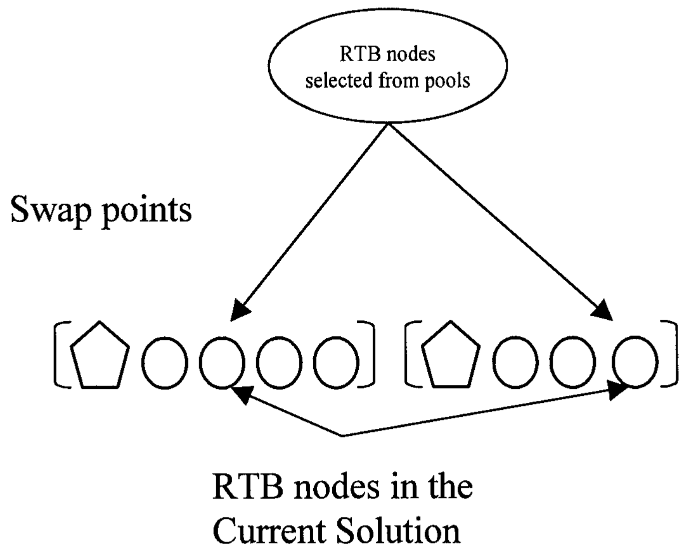

Figure 3.28: RTB Swap Move Generation. This figure illustration the swap points for an unassigned RTB node within the current solution. Assigned tankers are represented by pentagons and other nodes by circles.

RTB nodes in the current solution.

Move Evaluation

Once the generation of moves is completed, the evaluation of these moves begins. Initially, the first move evaluated is considered the best move found. All other moves are compared against the current best move found by using the objectives described in Section 3.3.3. Since the objective is hierarchical, the most important objective is evaluated first. If the current move is worse than the current best move, no further evaluations take place. If the current move is better, the remaining objectives are evaluated. If the current move is equal to the current best move, then the next most important objective in the heirarchy is evaluated. This pattern continues until the current move has

either been dropped or completely evaluated. The following sections describe the evaluation process of applying a move to the current solution in detail.

Evaluations not requiring time This section describes the objectives listed in Section 3.3.3 that are evaluated without the need for the timing of events. The first four objectives listed in Section 3.3.3 require only the specific partitioning and ordering of the waypoint nodes to the tankers as contained in the S_n representative. Each of the objective functions scans the solution resulting from the application of the current move upon the current solution. The unescorted RGs objective determines how many required waypoint node escort pairings do not occur within the tankers' assignments. The uncovered RG demand objective determines the number of waypoint nodes not found within the tankers' assignments. The misordered tanker services determines the number of waypoint nodes that violate the precedence conditions of the AFRP. The bad tanker assignments objective determines if a move creates a condition that is considered illogical. Additionally, the number of tankers used objective from Section 3.3.3 is determined by the number of cyclic partitions in the S_n representative. Each of these objectives may be evaluated without the determination of the specific timing of events.

Evaluations requiring time The remaining objectives require the specific timing of events in order to evaluate. Once the timing of events is determined, the evaluations of each of these objectives is straightforward; therefore, the remainder of this section will concentrate on the determination of the timing of events for a particular move being applied to the current solution.

The determination of the timing of events occurs through the application of the following three steps:

1. Determine each RG's flight path times regardless of tanker assignment
2. Determine each tanker's route times regardless of assigned RG times
3. Deconflict schedules determined in steps 1 and 2

The first step takes each individual RG and, based on its EDT with no delays, determines the arrival, service, and departure times at each waypoint node along its flight path as well as its destination arrival time. Similarly, the second step takes each individual tanker and, with no delays, determines the arrival, service, and departure times at each of its assigned waypoint nodes.

The third and final step in determining the timing of events for a solution deconflicts the schedules created in the first two steps through an iterative approach. This iterative approach involves two primary stages. The first stage considers each RG and determines the largest difference between its arrival at any waypoint node and the arrival of the tanker assigned to that waypoint node. For each RG whose largest difference is positive, the starting time of the RG is adjusted by the amount of this largest difference. The second stage considers each tanker and its associated route assignment. The difference between the tanker arrival at its first waypoint node and the associated RG arrival is determined. The tanker's schedule is adjusted to ensure that it either arrives at the same time as the RG or "orbits" until the RG arrives. Once the first waypoint node in the tanker's assignment is deconflicted, the

procedure continues with the next waypoint node in its assignment. This iterative approach continues until all nodes within the tanker's assignment have been deconflicted. When all tankers have been deconflicted, the deconflict step repeats. This iterative process continues until no conflicts are found or 10 iterations of the deconflict step have been completed. If 10 iterations are reached, the current candidate move is removed from consideration.

The choice of placing all "orbit" time in the tanker schedules is based on the capacity of the airframes that comprise the RGs, particularly fighters. For smaller aircraft, their fuel capacity minimizes their allowable orbit times waiting for a tanker to arrive. With the timing of events determined, the remaining objectives are evaluated.

Move Selection and Implementation

Once the best move is determined by the objective function, it is placed within the tabu structure. After this placement, the best move selected is implemented on the current solution, creating the next solution to be used in the tabu search iterative process.

3.3.6 Preprocessing

The description of the GTTS in the previous sections assumed that the RG WPTs were provided by an external source and were consistent, i.e., feasible (flyable) solutions could be found when those WPTs were used. As Section 4.2 illustrates, externally supplied WPTs are not necessarily consistent. To account for this possibility, a modified form of the GTTS, the GTTS Preprocessor (GTTSP), has been developed to determine consistent WPTs for a

single RG's flight path. Hence, the GTTSP is *also* an adaptive tabu search method developed specifically to find consistent active WPT node sets for a single RG. Reasons for using the GTTSP include:

- no externally supplied WPTs are available
- the provided WPTs are untested
- a comparison of the supplied WPTs against the GTTSP WPTs is desired

GTTSP requires the use of a set of candidate WPTs (nominally 100 nautical miles (NM) apart) throughout the selected RG's flight path. From this set of candidate WPTs, a subset will be selected as "active." This "active" subset of WPTs will define the actual WPTs for the RG's flight path in the AFRP to be solved by the GTTS.

The active subset of WPTs is selected based on the same objectives described in section 3.3.3 and using the same tanker resource configuration that will be used by the GTTS. The discussion of the GTTSP is divided into two sections, the no escort case and the escort case.

GTTSP No Escort Required

For both cases, the GTTSP generates WPT nodes in a slightly different way from the GTTS. For the no escort case, each candidate WPT is associated with a *single* candidate WPT node. Once the set of WPT nodes are available, an initial WPT node selection set for the individual RG is constructed by first selecting the "midway" WPT node along the RG flight path as represented in (3.4). (If the second term of (3.4) yields a noninteger amount, it

is truncated. The first "WPT node ID" $\equiv 1$ and the midway WPT node ID = (last WPT node ID - first WPT node ID) /2)

$$\text{(a tanker ID, midway WPT node ID)} \tag{3.4}$$

The amount of fuel required at this active WPT node is computed to be equal to the RG fuel usage to reach the midway WPT node's location. Depending on the consistency of this initial WPT node selection, one of the following two neighborhoods is employed:

- a Preprocessor Tanker Insert Move Neighborhood (PTKI) or
- a Preprocessor WPT Node Adjacent Swap Neighborhood (PAS).

If the current WPT node set is inconsistent, the PTKI is used. Three conditions can cause an initial WPT node set to be inconsistent:

- no tanker can fly to the midway point, provide service to the RG, and then return to its original beddown base without running out of fuel
- the RG can not fly to the midway WPT node without running out of fuel
- the RG can not fly from the midway WPT node to its destination base without running out of fuel

If any of these conditions exist, then the PTKI adds a WPT node to the current active set at the halfway point between the midway WPT node and either the first candidate WPT node or the last candidate WPT node

on the RG flight path. The new active WPT node is assigned to one of the remaining available tankers. Assuming that the halfway point between the first WPT node and the midway WPT node is chosen, the "new" solution would be represented by (3.5).

$$\left(\text{a tanker ID}, \frac{\text{(midway WPT node ID-first WPT node ID)}}{2}\right) \tag{3.5}$$
$$\text{(a tanker ID, midway WPT node ID)}$$

Once again, the GTTSP would determine the consistency of the current active WPT node set (as in (3.5)). If the active WPT node set is still inconsistent, the PTKI would be employed again. In this manner, the PTKI determines the cardinality of the active node set. Once a feasible solution is obtained through the iterative use of PTKI, PAS is employed.

PAS serves to iteratively improve the "active" node set generated by PTKI. It achieves this by taking each "active" node and generating a move that replaces it with either its predecessor or successor along the flight path. For example, if the PTKI generated solution is

$$(0, 50)(6, 75)$$

then the associated move neighborhood would allow the following four solutions to be reachable:

Shift 50 to predecessor	$(0, 49)(6, 75)$
Shift 50 to successor	$(0, 51)(6, 75)$
Shift 75 to predecessor	$(0, 50)(6, 74)$
Shift 75 to successor	$(0, 50)(6, 76)$

The best active WPT node set found by iteratively applying the PAS within the GTTSP then serves to define the flight path WPTs for the RG in the AFRP.

GTTSP Escort Required

When escort is required for a RG, WPT nodes are generated from the candidate WPTs in the following manner. If a WPT is not located over open water, then a *single* candidate WPT node is generated. However, if a WPT node is located over open water and the next adjacent WPT along the flight path is also over open water, then *two* candidate WPT nodes are generated similar to the splitting of WPTs described in Section 3.2.1. For the GTTSP, it is assumed that all WPT nodes requiring escort are connected. The *escort range* is a single contiguous interval starting at the first WPT escort node and ending at the last WPT escort node.

The initial solution is constructed using the first and last WPT nodes as represented in (3.6).

$$\text{(a tanker ID, first WPT node ID, last WPT node ID)} \tag{3.6}$$

The interpretation of this solution is as follows: the tanker flies to the first WPT node, provides service to the RG, escorts the RG to the last WPT node, provides service to the RG, and then returns to its beddown base. The fuel requirements at each of these "active" WPT nodes are determined by the GTTSP and, depending on consistency, either the PTKI or the PAS neighborhood is employed.

If the active WPT nodes set is inconsistent, the PTKI is used. The PTKI then determines the next active WPT node by considering the escort requirement of the RG. One of two cases can occur:

1. the midway WPT node is outside the escort range, or

2. the midway WPT node is within the escort range.

For case 1, two subcases can occur:

1a. the midway WPT node occurs before the start of the escort range or

1b. the midway WPT node occurs after the end of the escort range.

If subcase 1a occurs, the active WPT node set is represented as in (3.7).

(a tanker ID, first WPT node ID) (3.7)

(a tanker ID, midway WPT node ID, last WPT node)

The representation of (3.7) implies that one tanker flies to the first WPT node, services the RG, and then returns to its original beddown base. The other tanker now flies to the midway WPT node, services the RG, escorts the RG starting from the midway WPT node to the last WPT node, services the RG, and then returns to its original beddown base.

If subcase 1b occurs, the active WPT node set is represented as in (3.8).

(a tanker ID, first WPT node ID, midway WPT node ID) (3.8)

The representation of (3.8) implies that the RG is no longer escorted over its entire flight path. Rather, the RG is escorted over just the first half of its flight path before its assigned tanker leaves.

For case 2, the current tanker's assignment is split at the midway WPT node as represented by (3.9).

(a tanker ID, first WPT node ID, midway WPT node ID-1)

(a tanker ID, midway WPT node ID, last WPT node ID) (3.9)

(3.9) implies that one tanker flies to the first WPT node, services the RG, escorts the RG to the midway WPT node, services the RG, and returns to its beddown base. In a similar fashion, the other tanker provides escort from the midway WPT node to the last WPT node.

(3.7), (3.8), and (3.9) all demonstrate solutions derived from separating a tanker that serves escort arcs within its current assignment. However, as the search progresses, tankers that serve WPT nodes outside the escort range can be generated, and, in these cases, the PTKI follows (3.4) and (3.5). After consistency is achieved, the PAS is applied.

The PAS is similar to that described in Section 3.3.6 with the exception that escort pairings are maintained. To demonstrate this neighborhood, assume a RG that generated 100 candidate WPTs nodes numbered from {118, 119,...217} with escort pairings occurring between the nodes {142, 143,..., 175} (escort pairs being two-tuples starting with the pair (142,143) and ending with (174,175)). Suppose the GTTSP has obtained a consistent set of active WPTs nodes represented by (3.10)

$$(0, 125)(1, 142, 167)(2, 168, 180)(3, 200) \tag{3.10}$$

The PAS moves allow the active WPT nodes to be replaced with either their predecessor or successor along the flight path (described in Section 3.3.6). However, if the predecessor is the end node of an escort pair, then its predecessor (the start of the escort pair) becomes the allowable move. Likewise, if the successor is the start of an escort pair, then its successor (the end of the escort pair) becomes the allowable move. Additionally, if two tankers contain

consecutive WPT nodes falling within the escort node range (i.e., tankers 1 and 2), both tankers' assignments are adjusted to retain escort arc coverage Using (3.10), the following moves are generated:

Shift 125 to predecessor	$(0,124)(1,142,167)(2,168,180)(3,200)$
Shift 125 to successor	$(0,126)(1,142,167)(2,168,180)(3,200)$
Shift 142 to predecessor	$(0,125)(1,141,167)(2,168,180)(3,200)$
Shift 142 to succ. of succ.	$(0,125)(1,144,167)(2,168,180)(3,200)$
Shift 167 to pred. of pred.	$(0,125)(1,142,165)(2,166,180)(3,200)$
Shift 167 to succ. of succ.	$(0,125)(1,142,169)(2,170,180)(3,200)$
Shift 180 to predecessor	$(0,125)(1,142,167)(2,168,179)(3,200)$
Shift 180 to successor	$(0,125)(1,142,167)(2,168,181)(3,200)$
Shift 200 to predecessor	$(0,125)(1,142,167)(2,168,180)(3,199)$
Shift 200 to successor	$(0,125)(1,142,167)(2,168,179)(3,201)$

The PAS is applied for all iterations once feasibility is reached. At this point, assume that the construction neighborhood and the "shift the ends" neighborhood have been applied to the current RG. Suppose the active WPT node set is represented by (3.11)

$$(0,125)(1,130,160)(2,161,175)(3,204) \tag{3.11}$$

The following interpretation of the solution can be made. In (3.11), Tanker0 supplies fuel to the RG at WPT Node125 and then returns to its beddown base. Tanker3, likewise, supplies fuel at Node204 and then returns to its beddown bases. Tankers 1 and 2 have a slightly different assignment. Tanker1 supplies fuel at Node130, escorts the RG to Node160 where it supplies more fuel, and then returns to its beddown base. Likewise, Tanker2 supplies fuel at Node161, escorts the RG to Node175 where it supplies more fuel, and then returns to its base. Results from using this approach are presented in section 4.3.

This chapter has provided a detailed description of the GTTS approach to the AFRP that was used in this research. The next chapter discusses how this methodology was applied to example instances of the AFRP.

Chapter 4

Applying Group Theoretic Tabu Search to the Aerial Fleet Refueling Problem

Chapter 3 discussed the construction and implementation of a GTTS solution approach to the AFRP. This chapter reports the results of applying this method to several examples of the AFRP.

4.1 Simple Example

To facilitate an understanding of the many complex interactions that embody the solution to an AFRP, we limit our initial presentation of experimental results to a small problem. Consider three RGs that need to deploy from the continental US to Saudi Arabia. Table 4.1 provides the deployment details of the RGs and Figures 4.1, 4.2 and 4.3 display the flight paths.

Appendix A contains the actual locations of the origin and destination bases.

RG ID	AC Type	No. AC	Origin	Destination	EDT	RDT
0	F15	6	KLFI	OERY	8	40
1	F15	6	KLFI	OERY	0	40
2	F117	6	KHMN	OEDR	0	40

Table 4.1: Simple Example RG Deployment

Figure 4.1: RG0 Flight Path. This figure shows the flight path associated with RG0. RG0 starts at Langley AFB, VA (pentagon) and travels east across the Atlantic to Riyadh Air Base. Each circle represents a WPT where the RG refuelings are to take place (until the final destination). The WPT node IDs are shown next to the WPT. The number above the WPT represents a WPT node with fuel demand. The number below the WPT represents the beginning of a required escort. The yellow pentagons represent the active tanker bases.

Figure 4.2: RG1 Flight Path. This figure shows the flight path associated with RG1. RG1 starts at Langley AFB, VA (pentagon) and travels east across the Atlantic to Riyadh Air Base. Each circle represents a WPT where the RG refuelings are to take place (until the final destination). The WPT node IDs are shown next to the WPT. The number above the WPT represents a WPT node with fuel demand. The number below the WPT represents the beginning of a required escort. The yellow pentagons represent the active tanker bases.

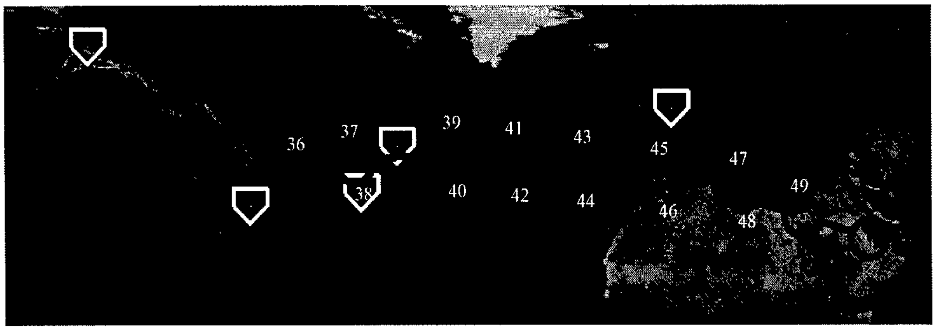

Figure 4.3: RG2 Flight Path. This figure shows the flight path associated with RG2. RG2 starts at Holloman AFB, NM (white pentagon) and travels east across the Atlantic to King Abdul Aziz Air Base. Each circle represents a WPT where the RG refuelings are supposed to take place (until the final destination). The yellow pentagons represent the active tanker bases.

These RGs may be serviced by any of 18 KC-135R tankers located at:

- Bangor International Airport, ME (KBGR) - 6 tankers
- Mildenhall AB, UK (EGUN) - 6 tankers
- Seymour-Johnson AFB, NC (KGSB) -3 tankers
- Eielson AFB, AK (PAEI) - 3 tankers

These bases are represented by the yellow pentagons in Figures 4.1, 4.2 and 4.3. All tankers are available to take-off immediately.

4.1.1 Assumptions

For this small example, a number of assumptions have been made. Specifically,

- the WPT locations and fuel demands have been fixed *a priori*,

- the WPT refuelings assume that the RG does not need to change speed or altitude for the refuelings to take place,
- the WPT refuelings do not reflect movement along the flight path during the service time, and
- the RG flight paths and tanker assignments do not consider Crew-Duty-Day restrictions.

4.1.2 Results

The results shown in the following section are based on a 30 minute run of the GTTS on an AMD Athlon 950 MHz machine with 256 MB RAM. The best solution was found at iteration 369 (out of 430) in just under 25 minutes. A solution snapshot plot of the search process can be seen in Figure 4.4. As mentioned in Section 3.2.1, the tankers are assigned consecutive numbers 0 to 17 and RGs 0, 1, and 2 are assigned, respectively, WPT nodes 18-26, 27-35, and 36-49. Nodes numbered higher than 49 represent tanker return to base nodes.

The permutation representation of the best solution is:

(0,*18*)(2,**27**,58,*19,20*,75,37,38,39)(3,40,41)(6,**32,33**,56,*25,26*)
(7,**30,31**,55,44,45)(8,46,47)(9,*23,24*,59,48,49)(10,**34,35**)
(11,*21,22*,57,42,43)(12,**28,29**,53,36)

In this representation, the normal font (0) represents tankers, the italicized numbers (*18*) represent waypoints for RG0, the bold numbers (**27**) RG1,

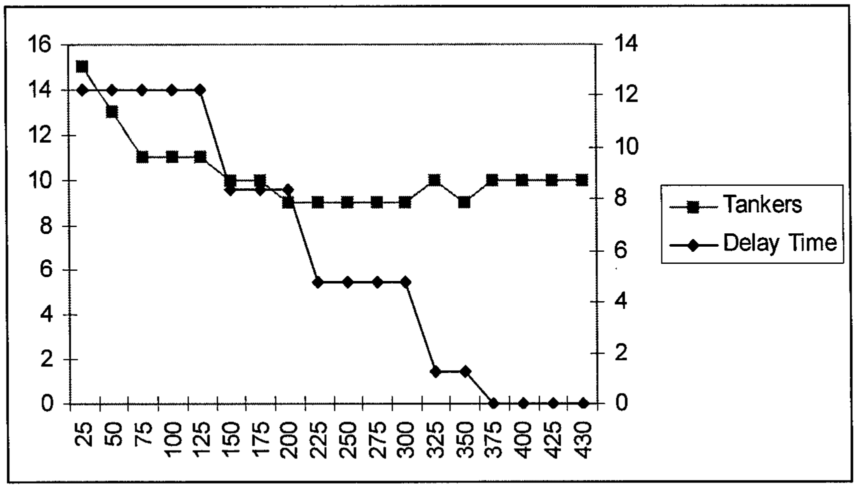

Figure 4.4: Solution Snapshot Plot. This figure shows two objective function values (Delay Time and Tankers) for the best solution at every 25 iterations of the search process. The left axis represents the number of tankers and the right axis represents the amount of delay at WPT nodes (in hours).

the typewriter numbers (`37`) RG2, and the large numbers (53) return to base nodes.

Translating this representation yields (for a complete description including the timing of events, see Appendix B):

(0,*18*) $\Longrightarrow$ Tanker0 located at KBGR provides fuel for RG0 at Node18 and returns to KBGR.

(2,**27**,58,*19*,*20*,75,`37`,`38`,`39`) $\Longrightarrow$ Tanker2 located at KBGR provides fuel for RG1 at Node27, returns to KBGR, recovers for a period of 4 hours, meets RG0 at Node19 and escorts RG0 to Node20 where it provides fuel, returns to KBGR, recovers for a period of 4 hours, provides fuel for RG2 at Node37, escorts RG2 from Node38 to Node39 where it provides fuel, and finally returns back to KBGR.

(3,40,41) $\Longrightarrow$ Tanker3 located at KBGR meets RG2 at Node40, escorts RG2 to Node41 where it provides fuel, and returns to KBGR.

(6,**32,33**,56,*25,26*) $\Longrightarrow$ Tanker6 located at EGUN meets RG1 at Node32, escorts RG1 to Node33 where it provides fuel, returns to EGUN, recovers for a period of 4 hours, meets RG0 at Node25, escorts RG0 to Node26 where it provides fuel, and returns to EGUN.

(7,**30,31**,55,44,45) $\Longrightarrow$ Tanker7 located at EGUN meets RG1 at Node30, escorts RG1 to Node31 where it provides fuel, returns to EGUN, recovers for a period of 4 hours, meets RG2 at Node44, escorts RG2 to Node45, and returns to EGUN.

(8,46,47) $\Longrightarrow$ Tanker8 located at EGUN meets RG2 at Node46, escorts RG2 to Node47 where it provides fuel, and returns to EGUN.

(9,*23,24*,59,48,49) $\Longrightarrow$ Tanker9 located at EGUN meets RG0 at Node23, escorts RG0 to Node24 where it provides fuel, returns to EGUN, recovers for a period of 4 hours, meets RG2 at Node48, escorts RG2 to Node49 where it provides fuel, and returns to EGUN.

(10,**34,35**) $\Longrightarrow$ Tanker10 located at KGSB meets RG1 at Node34, escorts RG1 to Node35 where it provides fuel, and returns to KGSB.

(11,*21,22*,57,42,43) $\Longrightarrow$ Tanker11 located at KGSB meets RG0 at Node21, escorts RG0 to Node22 where it provides fuel, returns to EGUN, recovers for a period of 4 hours, meets RG2 at Node42, escorts RG2 to Node43 where it provides fuel, and returns to KGSB.

(12,**28,29**,53,36) $\Longrightarrow$ Tanker12 located at KGSB meets RG1 at Node28,

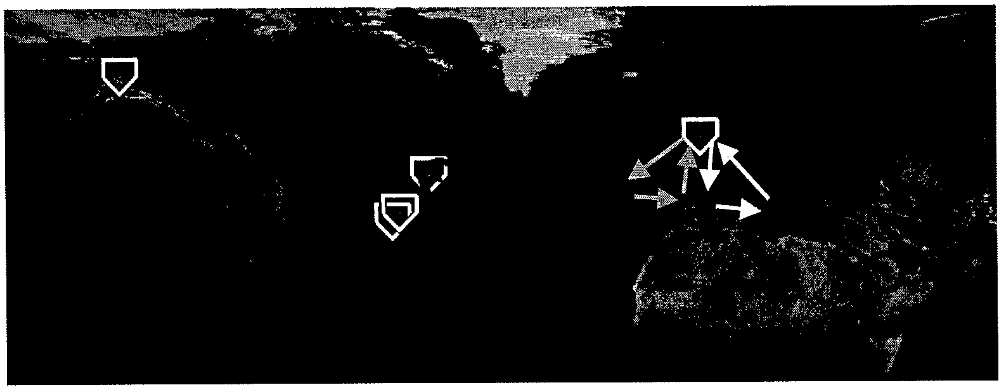

Figure 4.5: Simple Example's Tanker6 Assignment. This figure shows how Tanker6 takes off from EGUN, services RG1, and then returns to EGUN (shown in green). After a service period delay, the tanker takes off from EGUN, services RG0, and returns to EGUN (shown in yellow). This figure shows how a tanker is allowed to make multiple trips from its home base. RG0 and RG1 use the same flight path.

escorts RG1 to Node29 where it provides fuel, returns to KBGR, recovers for a period of 4 hours, provides fuel for RG2 at Node36, and then returns to KGSB.

Figures 4.5 and 4.6 provide a graphical view of the activities of tankers 6 and 12, respectively. These tankers are chosen because they represent the two types of return to base nodes that were introduced into the solution. Tanker6 returns to its original beddown base while Tanker12 refuels at a different base before finishing at its original beddown base.

Figure 4.7 depicts the assignment of all tankers to RGs for the simple example.

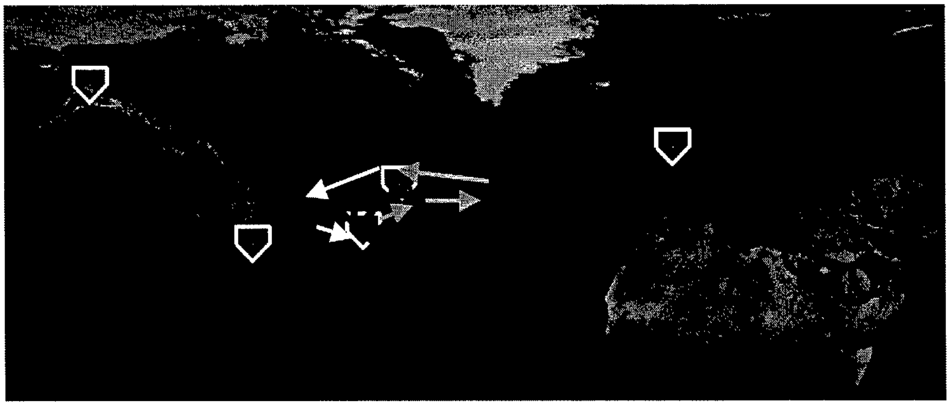

Figure 4.6: Simple Example's Tanker12 Assignment. This figure shows how Tanker12 takes off from KGSB, services RG1, and then proceeds for refueling to KBGR (shown in green). After a service period delay, the tanker takes off from KBGR, services RG2, and returns to KGSB (shown in yellow). This figure shows how a tanker is allowed to make multiple trips from varying base locations.

4.2 Benchmark Problem

Capehart (2000) developed the TAP Tool and attempted several AFRP deployment problems. For the purposes of comparison, Capehart's Middle East Deployment has been selected. This deployment (shown in Table 4.2) consists of 9 RGs and will serve as the baseline deployment for the following comparison cases:

1. As close to the benchmark problem as possible,
2. Relaxing some of the escort requirements of the benchmark problem,
3. Redefining the WPTs to create a feasible problem for the RGs of the benchmark problem

Appendix A contains detail on the actual locations listed in the origin

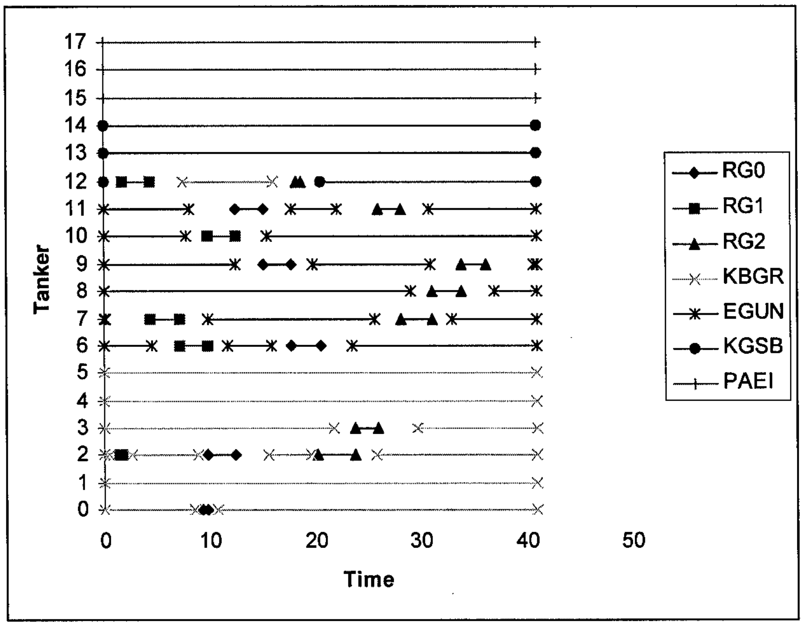

Figure 4.7: Tanker Assignments for Simple Example. This figure plots, over time, the tankers that serve each of the RGs in the simple example. Active tankers contain intervals along the time axis. These intervals include idle time at a base (KBGR, EGUN, KGSB, PAEI) or active periods servicing and/or escorting a RG. For RGs, short intervals represent the service time of the tanker at a WPT. Long intervals for RGs represent the escort time between WPTs and the service time at the end WPT.

RG ID	AC Type	No. AC	Origin	Destination	EDT	RDT
0	F117	2	KHMN	OEDR	0	96
1	A10	6	PAEI	OEDR	0	96
2	F15	6	KLFI	OEDR	0	96
3	F16	6	KSSC	OEKM	0	96
4	F15	6	KLFI	OEKM	0	96
5	B1	1	KMUO	FJDG	0	96
6	B1	1	KRCA	FJDG	0	96
7	B52	1	KDYS	FJDG	0	96
8	B52	1	KMIB	FJDG	0	96

Table 4.2: Benchmark Middle East Deployment

and destination columns. The RGs in Table 4.2 may be served by tankers located at the following bases:

- KBGR - 10 tankers
- EGUN - 30 tankers
- KGSB -10 tankers
- PAEI - 10 tankers

Figure 4.8 depicts the RGs' flight paths, the WPTs selected by Capehart's TAP Tool, and the tanker base locations.

4.2.1 Assumptions for the Benchmark

The assumptions for this problem include those mentioned in Section 4.1.1 as well as the following:

- the WPTs generated by Capehart (the circles in Figure 4.8) are used by this research for comparison purposes only. However, these WPTs do not allow the bombers to complete their missions before running out of fuel. (A method was used that incorrectly computed the location and number of refuelings required.) In Section 4.3, an alternate set of WPTs is generated by the GTTSP to ensure that all RG flight paths are feasible for both RGs and tankers.
- RGs requiring escort are escorted starting at their *first* WPT, regardless of whether or not the WPT is located over open water.

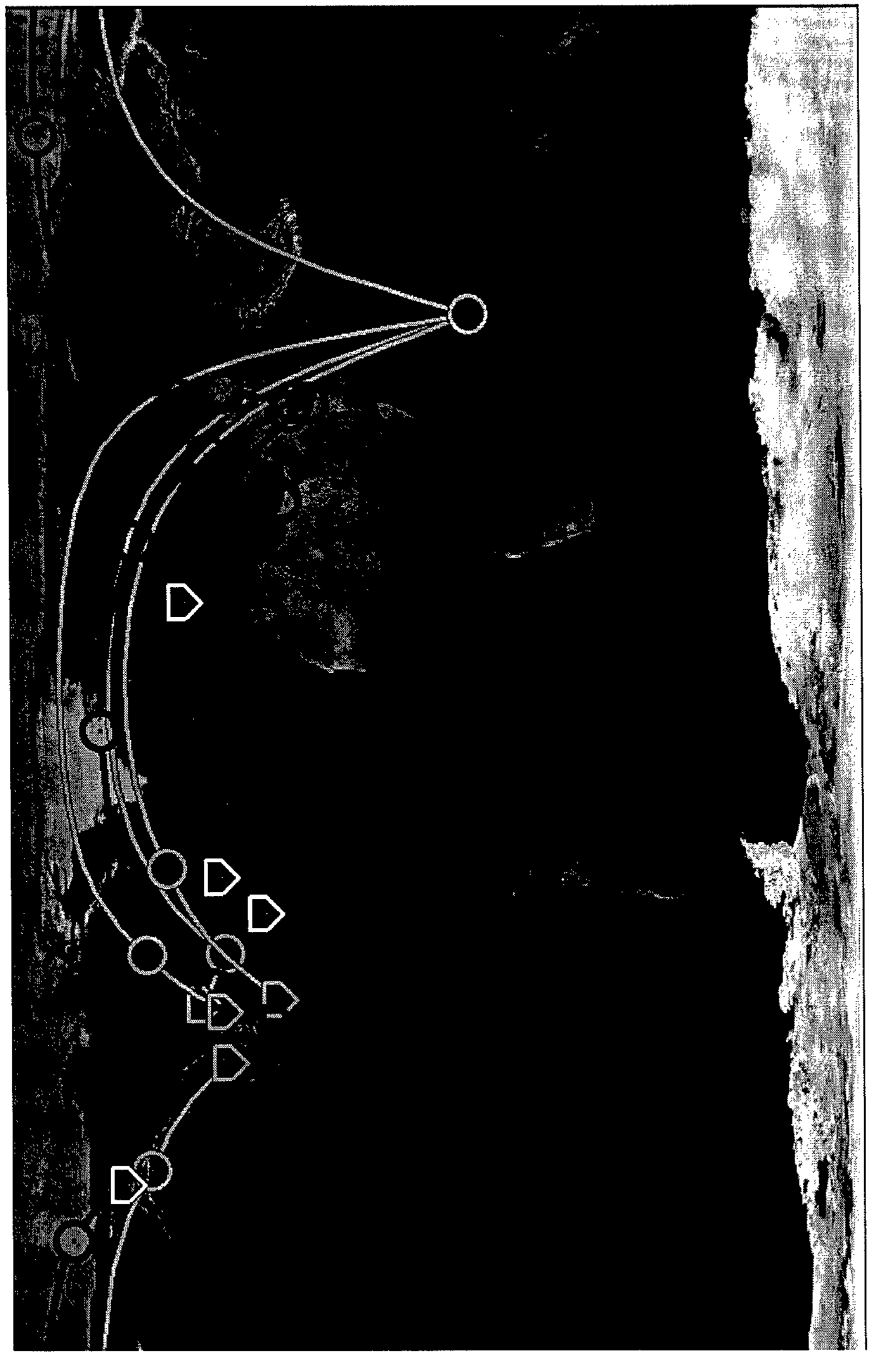

Figure 4.8: Middle East Deployment Benchmark. This figure displays the flight paths of the RGs in Capehart's Middle East Deployment. The red lines represent RGs that require escort (fighters) and the green lines represent RGs that do not require escort (bombers). Each WPT is represented by a circle. The yellow (and white) pentagons represent beddown bases for the available tankers.

4.2.2 Differences in TAP Tool and GTTS

Capehart's TAP Tool and the GTTS approach, as applied to this benchmark, contain a number of differences. These differences include:

- Last Leg Escort: Capehart escorts RGs of "light" aircraft from their first WPT to their destination base while the model in this research escorts these RGs from their first WPT to their last WPT before the final destination base.

- WPT Node Assignment: Capehart requires that a tanker immediately return to its beddown base whenever it completes service at an assigned waypoint. This research allows multiple WPT Nodes to be served by a single tanker if it has the capability to do so.

4.2.3 Results for the Benchmark Problem

Table 4.3 compares Capehart's best solution with the best solution found by GTTS.

The TAP Tool column gives the best solution found during 6 runs with different tabu tenures. Each run was for 90 minutes using an Intel Pentium II 350 MHz machine with 64 MB RAM. The GTTS results are from a single run of 30 minutes using an AMD Athlon 950 MHz machine with 256 MB

	TAP Tool	**GTTS**
No. Tankers	24	23
Total Tanker Distance	215204	106227
Latest RG Arrival	69.1	69.2

Table 4.3: Comparison Between TAP Tool and GTTS (Case 1)

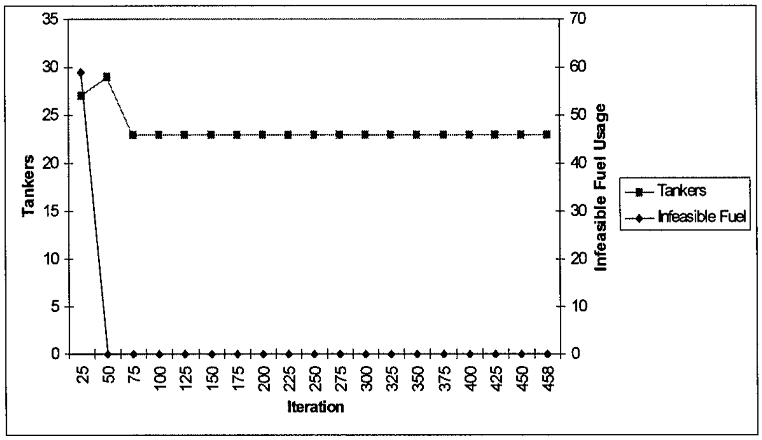

Figure 4.9: Solution Snapshot Plot for Benchmark Comparison (Case 1). This figure shows the objective function values (Infeasible Fuel Usage and Tankers) for the best solution at every 25 iterations of the GTTS search process on the benchmark problem with the original escort restrictions. The left axis represents the number of tankers and the right axis represents the amount of infeasible fuel usage (in 1000's of pounds).

RAM with the time to best solution occurring at 2 minutes 32 seconds into the search. Figure 4.9 shows a solution snapshot plot of some of the results from the search process.

The difference between TAP Tool and GTTS in the total tanker distance shown occurs because Capehart requires the tankers to return to their beddown bases. A more realistic approach implemented in GTTS allows the tankers to land at other active tanker bases during the deployment. This eases the escort requirement and provides opportunity for better assignments of the tankers to be found. However, Capehart's stringent requirement that RGs of "light" aircraft be escorted for most of their flight path severely hampers the effectiveness of tanker relocation within the GTTS. In fact, if the escort requirement is limited to the portions of the flight paths where there is open

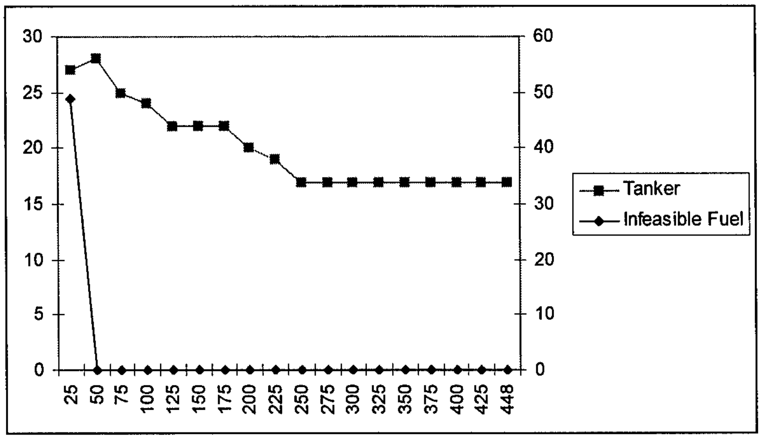

Figure 4.10: Solution Snapshot Plot for Benchmark Comparison (Case 2). This figure shows the objective function values (Infeasible Fuel Usage and Tankers) for the best solution at every 25 iterations of the GTTS search process on the benchmark problem with the original escort restrictions. The left axis represents the number of tankers and the right axis represents the amount of infeasible fuel usage by tankers (in 1000's of pounds).

sea between WPTs, the GTTS finds a markedly superior best solution (in a single run of 30 minutes) in about 12 minutes as shown in Table 4.4. Figure 4.10 provides a snapshot of the search progress for this problem.

Clearly, the escort requirement plays a significant role in determining the tanker requirement. However, the results shown in Table 4.4 are unrealistic since the bombers still run out of fuel before completing their missions because of the flawed WPTs. Section 4.3 discusses how the GTTSP was used

	GTTS
No. Tankers	17
Total Tanker Distance	118062
Latest RG Arrival	64.0

Table 4.4: GTTS Results Using Realistic Escorts (Case 2)

to determine a feasible placement of the WPTs for Capehart's Middle East Deployment.

4.3 GTTS Construction of the Benchmark Problem

As previously mentioned, the WPTs for Capehart's Middle East Deployment are flawed. To overcome this problem, the GTTS Preprocessor (GTTSP) was used to determine an excellent feasible placement of the WPTs along the given flight path of the RGs.

For each RG, a candidate set of WPTs was generated. This set consisted of points spaced 100 NM apart along the great circle route of the RG's flight path. From this WPT candidate set, the GTTS was run using tankers based at the same locations as the tanker bases of the benchmark problem. The WPTs selected by this process provide an excellent set of consistent WPTs (for both the tankers *and* the RGs) to use in the actual deployment problem. Figure 4.11 displays the candidate WPTs for the F117 RG coming from Holloman AFB, NM (KHMN) and the B52 RG coming from Minot AFB, ND (KMIB). Figure 4.12 displays the WPTs selected for each of these two RGs. In a similar manner, all of the other RGs' WPTs were selected. Figure 4.13 gives the complete set of new WPTs.

With this new set of WPTs, Capehart's Middle East Deployment was again solved. The GTTS results with feasible WPTs and proper escort requirements produced the best solution (with a single run of 30 minutes) given in Table 4.5. The best solution was found in just under 30 minutes. This solution required 5 more tankers (bad), but no tankers or RGs crashed from

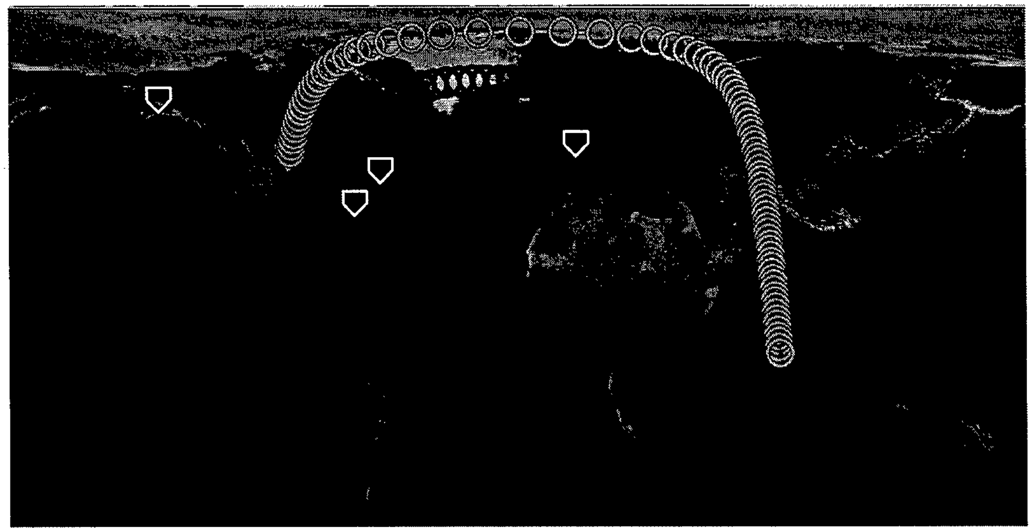

Figure 4.11: Initial WPT Candidates for GTTS Pre-run. This figure displays candidate WPTs for the F117s (in red) and the B52 (in green). The pentagons identify active tanker bases.

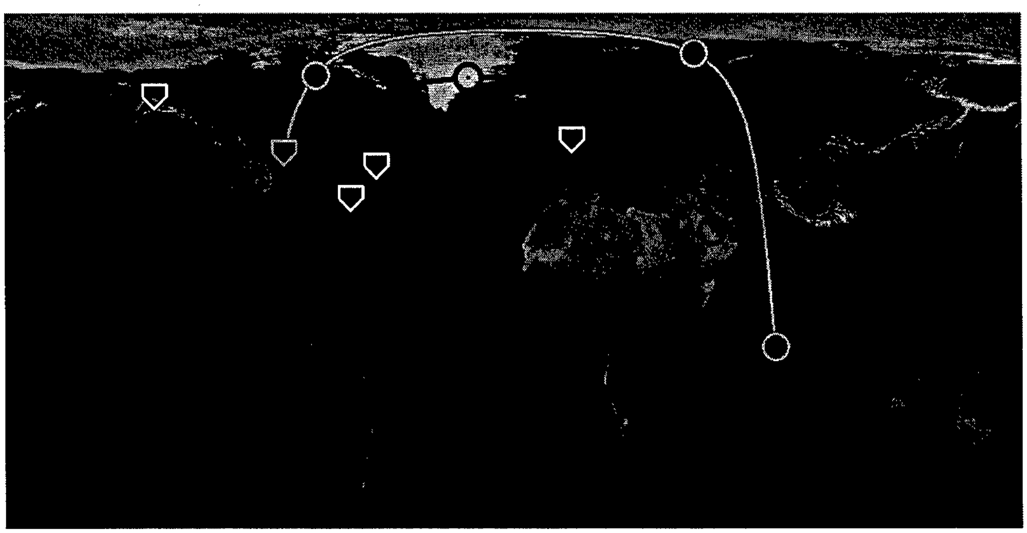

Figure 4.12: Selected WPTs from GTTS Pre-run. This figure shows the WPTs selected by the GTTS pre-run for use in the GTTS full model.

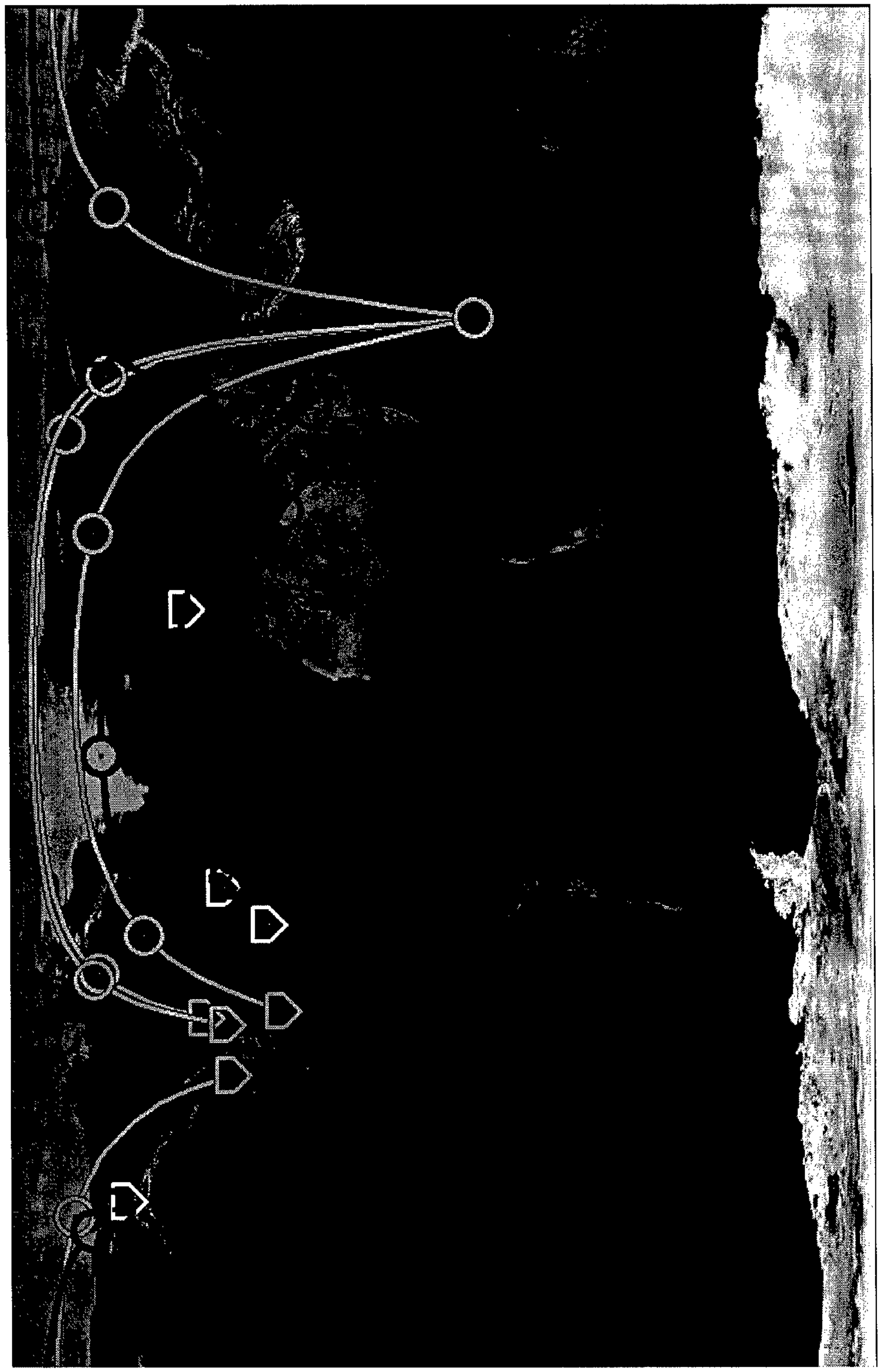

Figure 4.13: GTTS Middle East Deployment. This figure shows the GTTS selected WPTs for all the RGs in the Benchmark Comparison (Case 3).

	GTTS
No. Tankers	22
Total Tanker Distance	112864
Latest RG Arrival	60.5

Table 4.5: GTTS Reformulated Benchmark Results (Case 3)

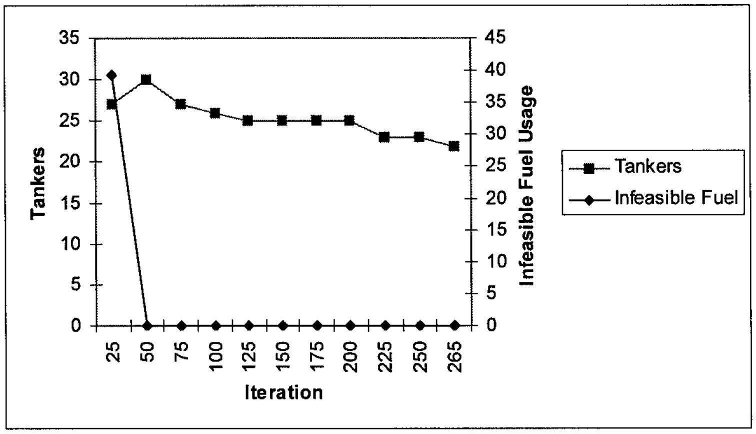

Figure 4.14: Solution Snapshot Plot for Benchmark Comparison (Case 3). This figure shows the objective function values (Infeasible Fuel Usage and Tankers) for the best solution at every 25 iterations of the GTTS search process on the benchmark problem (case 3). The left axis represents the number of tankers and the right axis represents the amount of infeasible fuel usage by tankers (in 1000's of pounds).

lack of fuel (GOOD!). Further, the tanker travel and the latest RG arrival are also improved. Figure 4.14 provides a snapshot view of the search progress for this problem.

Clearly, the proper bomber WPT placement affects the solution. Up to this point, this research has used Capehart's Middle East Deployment as a benchmark. The next section considers a deployment of practical size in terms of the number of RGs and tankers involved.

4.4 A Typical Middle East Deployment

This section describes the results of GTTS against a deployment of 99 aircraft in 26 RGs serviced by 120 tankers. Table 4.6 lists the composition of the RGs and Table 4.7 lists the beddown bases for the tankers.

The GTTSP was used for each of the RGs to ensure that all WPT placements were feasible. Once the WPTs were determined, GTTS was run for 12 hours on an AMD Athlon 950 MHz machine with 256 MB RAM. The first feasible solution consisting of 116 tankers was found about 76 minutes into the process. The best solution, given in Table 4.8 was found about 2 hours and 16 minutes into the process. Figure 4.15 provides a solution snapshot plot of the search progression.

This chapter started with a simple example to guide the interpretation of the GTTS results. It then showed that the GTTS was able to find superior solutions to Capehart's Middle East Deployment. The chapter then demonstrated the capability of the GTTSP to determine feasible WPTs for use in the GTTS. Finally, the chapter provided the results of using the GTTSP in conjunction with the GTTS to solve a more typical and difficult Middle East Deployment. Chapter 5 presents the concluding remarks pertaining to this dissertation.

RG ID	AC Type	No. AC	Origin	Destination	EDT	RDT
0	A10	6	KPOB	OEDR	0	168
1	A10	6	KPOB	OEDR	0	168
2	B1	1	KRCA	FJDG	0	168
3	B1	1	KRCA	FJDG	0	168
4	B2	1	KSZL	FJDG	0	168
5	B2	1	KSZL	FJDG	0	168
6	B52	1	KBAD	FJDG	0	168
7	B52	1	KBAD	FJDG	0	168
8	E3	2	KTIK	OERY	0	168
9	E3	2	KTIK	OERY	0	168
10	E8	1	KWRB	FJDG	0	168
11	E8	1	KWRB	FJDG	0	168
12	E8	1	KWRB	FJDG	0	168
13	F117	6	KHMN	OEDR	0	168
14	F15	6	KLFI	OEDR	0	168
15	F15	6	KLFI	OEDR	0	168
16	F15	6	KLFI	OEKM	0	168
17	F15	6	KLFI	OEKM	0	168
18	F15	6	KLFI	OERY	0	168
19	F15	6	KLFI	OERY	0	168
20	F15E	4	KGSB	OEKM	0	168
21	F15E	4	KGSB	OEKM	0	168
22	F16	6	KSSC	OEDR	0	168
23	F16	6	KSSC	OEDR	0	168
24	F16	6	KSSC	OEKM	0	168
25	F16	6	KSSC	OEKM	0	168

Table 4.6: A More Typical Middle East Deployment

Beddown Base	#Tankers
KBGR	20
EGUN	20
KGSB	20
PAEI	20
LPLA	20
OERY	20

Table 4.7: Beddown Bases for Typical Middle East Deployment

	GTTS
No. Tankers	95
Total Tanker Distance	326968
Latest RG Arrival (in hours)	55.0

Table 4.8: Typical Middle East Deployment Results

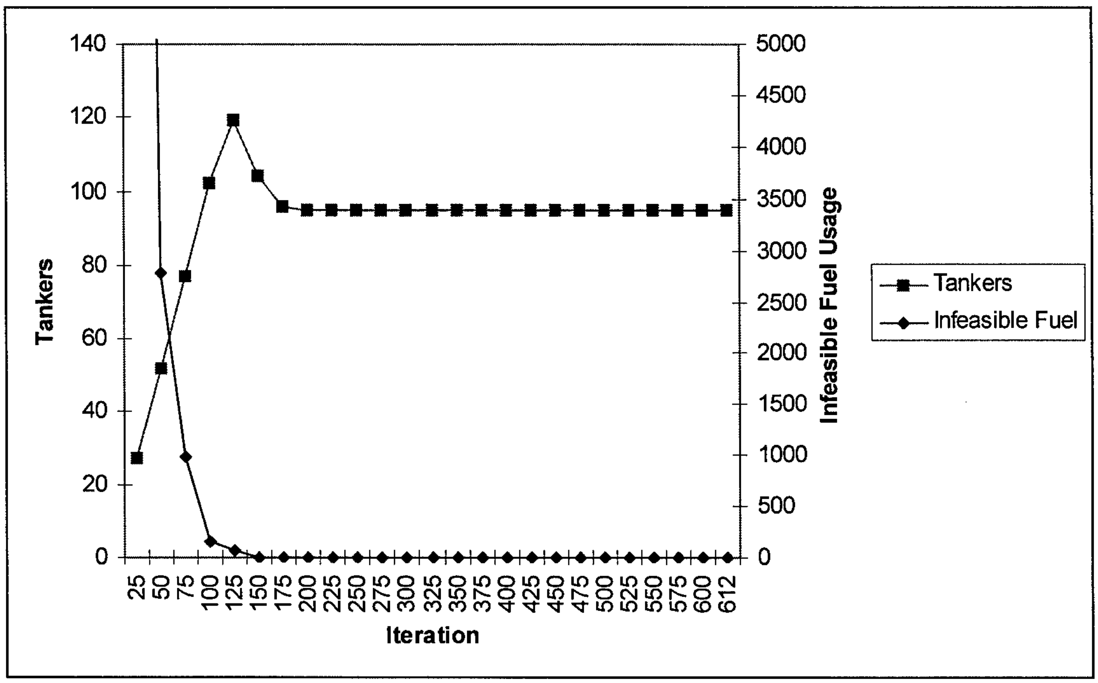

Figure 4.15: Solution Snapshot Plot for Typical Middle East Deployment. This figure shows the objective function values (Infeasible Fuel Usage and Tankers) at every 25 iterations of the GTTS search process. The left axis represents the number of tankers and the right axis represents the amount of infeasible fuel usage by RGs and tankers (in 1000's of pounds).

Chapter 5

Concluding Remarks

This chapter details the major contributions of this dissertation and suggests areas for future research and for extensions to other important AMC problem arenas.

5.1 Major Contributions

This research has yielded the following major contributions:

- a reusable, portable code for implementing and applying group theory to P|O combinatorial problems has been created
- the effectiveness of using dynamic search methodologies has been shown for the AFRP
- a very effective solution methodology for the AFRP has been developed

Appendices C & D briefly describe the Java language based approach to implementing S_n that was developed as part of this research effort. This implementation served as the foundation for this research, but it can be applied to any P|O combinatorial problem. The SymmetricGroup class described

in Appendix D inherently captures the partitioning of P|O problems with its cyclic structures and the ordering of P|O problems by the arrangement within the cycles. Within this class, built-in methods allow the user to access the standard group actions of multiplication and conjugation which allow a solution to be transformed into another solution. With these standard operations, the SymmetricGroup class is easily extended to any heuristic approach that makes use of rearrangement and insertion neighborhoods.

The use of dynamic neighborhood selection allows the search process to generate neighborhoods as the search progresses. This allows the GTTS to exploit current search space topology in more efficient ways.

5.2 Suggestions for Future Research

This section suggests areas where additional investigations would enhance the efficiency of the GTTS technique by enhancing both the Java-based implementation of group theory within the GTTS and the tabu search approach used within the GTTS.

5.2.1 Efficiency of the Java Group Theory Code

While the SymmetricGroup class (described in Appendix D) performs its duties admirably, there are computational improvements that could be made within the methods and data structures used by this class. An in-depth analysis of the code would provide opportunities for faster manipulation of the data structures.

5.2.2 Efficiency of the Tabu Search Code

The current GTTS uses a prototypical heuristic to assign all WPT nodes to a single tanker. Use of more sophisticated construction algorithms should produce better solutions sooner in the search. However, as was demonstrated, the tabu search clearly is able to overcome limitations imposed by the quality of the initial solution.

The current tabu search approach uses a fixed neighborhood "depth" restriction of 5 to limit the number of moves generated by the RI, the RS, and the EPI move neighborhoods. An investigation of the effects of adaptively varying this depth of the search as part of the dynamic search described in section 3.3.2 could produce better solutions in less time.

The current approach also uses a hierarchical objective function that places an infinite weight on values higher in the hierarchy. Implementing a weighting scheme combining the competing objectives could produce "smoother" surfaces yielding a more efficient search.

The current method uses an adaptive tabu search scheme that works well for the AFRP. However, implementation of a Reactive Tabu Search approach to the problem (Battiti & Tecchiolli 1994, Barnes & Carlton 1995, Nanry 1998, Zeisler 1999, Ryan, Bailey, Moore & Carlton 1999, O'Rourke 1999, Battiti & Bertossi 1999, Nanry & Barnes 2000, Harder 2000) would be an appropriate investigation. An investigation of more sophisticated tabu memory structures might also be beneficial.

5.3 Extensions to Other Important AMC Problems

This section discusses

- extensions of this research to other important AMC problems
- how the group theoretic framework can be applied across all partioning and ordering problems

5.3.1 Other AMC Problems

The AFRP is just one of a myriad of problems being addressed by AMC. In addition to the AFRP, AMC is addressing:

- refueling needs in intra-theater employment
- once tanker schedules have been determined, assigning aircrews to the tankers
- assigning the aircrews for RGs

Intra-theater employment

Once the deployment has taken place, aircraft will be used to perform various subsequent missions. While performing these missions, the aircraft will need to be refueled. The AFRP addressed in this research has the tankers flying to meet RGs, while employment problems typically have tankers "orbiting" at a single WPT.

Tanker Aircrew scheduling

Given an existing tanker schedule, aircrews must be assigned to the tankers. One way to ensure that the tanker schedules produced are feasible for the associated aircrews is to include crew duty day constraints in the AFRP. Alternately, the crew information could be included as another object within the AFRP structure. The aircrews could then explicitly be assigned within the search procedure itself.

RG Aircrew scheduling

Given an RG schedule, aircrews must be assigned to the RG aircraft. The current implementation of the AFRP does not explicitly account for RG aircrew duty day limitations. Several different methods can be used to address these limitations:

- include stops at base(s) as part of the flight path for an RG and require the RG to wait a period of time before being allowed to continue on.
- split the RG into several RGs with flight paths representing the beginning and ending of each day's allowable flight. Precedence relations between the RGs would then have to be reconciled.

5.3.2 Partitioning and Ordering Problems

This research applied the GTTS to the AFRP. However, the GTTS, or just the SymmetricGroup class, can be applied to any problem where the partitioning and ordering of elements is important. Colletti (1999) describes the use of S_n for representing the m-TSP. As the GTTS demonstrated, far more complicated

problems can be solved using S_n as the basic representation of solutions and for constructing varied move neighborhoods.

These move neighborhoods can be compactly represented using a the group actions of multiplication and conjugation. Any k-OrOpt move can be represented by three letters from S_n combined with right multiplication. The first of the three letters represent the beginning of the k-length pattern, the second represents the letter immediately following the end of the k-length pattern, and the third letter represents the letter to insertion point of the k-length pattern. Any m-letter rearrangement move can be represented by m-letters from S_n combined with conjugation. An excellent discussion of these rearrangement moves can be found in **?**. In fact, S_n facilitates the construction of very complex move neighborhoods in a compact representation through the use of conjugation and multiplication as described and demonstrated in Section 3.3.2 of this dissertation and in Combs (2001).

5.4 Summary

This research has

- created a reusable, portable code for implementing and applying group theory to P|O combinatorial problems
- demonstrated the effectiveness of using dynamic search methodologies for the AFRP
- yielded a very effective solution methodology for the AFRP

Computational results indicate that the GTTS is effective, providing good results with no tuning. The solutions found are superior to all known benchmark problems and the procedures are sufficiently robust to allow different objectives and constraints to be placed in the problem.

There are many avenues for future research. Such research will improve the efficiency of the Java-based GTTS by employing more sophisticated tabu search strategies and improved programming techniques. Future researchers should also be able to build on the foundation provided by this dissertation to construct highly effective approaches to other P|O combinatorial problems.

Appendix

Appendix A

Acronyms

AFRP Aerial Fleet Refueling Problem

AMC Air Mobility Command

AP Assignment Problem

CMARPS Combined Mating and Ranging Planning System

$m-$**CTSP** $m-$agent Cyclic Traveling Salesman Problem

EDT Earliest Departure Time

EGUN Mildenhall AB, UK

EPI Escort Pair Insert Move Neighborhood

FJDG Diego Garcia, British Indian Ocean Territory

GA Genetic Algorithm

GRASP Greedy Randomized Adaptive Search Procedure

GSICS Graphically Supported Interactive Control System

GTTS Group Theoretic Tabu Search

GTTSP GTTS Preprocessor

GVRP General Vehicle Routing Problem

ID Identification Number

JSSP Job Shop Scheduling Problem

KBAD Barksdale AFB, LA

KBGR Bangor International Airport, ME

KDYS Dyess AFB, TX

KGSB Seymour-Johnson AFB, NC

KHMN Holloman AFB, NM

KIAB McConnell AFB, KS

KLFI Langley AFB, VA

KMIB Minot AFB, ND

KMOT Minot International Airport, ND

KMUO Mountain Home AFB, ID

KPOB Pope AFB, NC

KRCA Ellsworth AFB, SD

KRDR Grand Forks AFB, ND

KSSC Shaw AFB, SC

KSKA Fairchild AFB, WA

KSZL Whiteman AFB, MO

KTIK Tinker AFB, OK

KWRB Warner-Robbins AFB, GA

LOC location

LPLA Lajes Field, Azores, Portugal

MCGI Mapping, Charting, Geodesy, and Imagery

MOG maximum on ground

NM nautical miles

OEDR King Abdul Aziz AB, Saudi Arabia

OEKM King Khalid AB, Saudi Arabia

OERY Riyadh AB, Saudi Arabia

OOP Object-Oriented Programming

OOPL OOP Language

P|O Partitioning and Ordering Problems

PAED Elmendorf AFB, AK

PAEI Eielson AFB, AK

PAS Preprocessor WPT Node Adjacent Swap Neighborhood

PGUA Andersen AFB, Guam

PTKI Preprocessor Tanker Insert Move Neighborhood

QLT Quick-Look Tool

RDT Required Delivery Time

RG Receiver Group

RI Restricted Insert Move Neighborhood

RKSO Osan AB, Republic of Korea

RODN Kadena AB, Japan

RS Restricted Swap Move Neighborhood

RTB Return To Base

RTBD Return To Base Delete Move Neighborhood

RTBI Return To Base Insert Move Neighborhood

RTBS Return To Base Swap Move Neighborhood

SCP Set-Covering Problem

SGforTS Symmetric Group for Tabu Search

S_n Symmetric Group on n-letters

TAP Tanker Assignment Planning

TKI Tanker Insert Move Neighborhood

TKS Tanker Swap Move Neighborhood

TMARP Tanker Mating and Ranging Program

TPFDD Time Phased Force Deployment Document

TS Tabu Search

TSP Traveling Salesman Problem

TSSAS TPFDD Sizing, Sourcing, and Analysis System

USAF United States Air Force

VRP Vehicle Routing Problem

WPT waypoint

Appendix B

Detailed Description of Simple Example

B.1 Computational Statistics

Iteration of Best Solution: 369

Total solve time before best: 1478110 milliseconds

Total time spent on 430 iterations was: 1726196 milliseconds

Best found in: 1537902 milliseconds

Total time spent: 1801661 (in milliseconds)

B.2 Objective Function Value

The following were the values associated with the hierarchical objective function:

Number of uncovered escort arcs	0
Number of uncovered RG demand nodes	0
Number of precedence pairs misordered	0
Number of bad tanker assignments	0
Amount of infeasible fuel usage	0.0
Amount of delay time	0.0
Amount of RG late arrivals	0.0
Overflow of MOG	0
Number of tankers used	10
Flying time (in hours) of tankers used	266.811
Distance traveled by tankers	48126.352
Amount of fuel used by tankers	891.625
Amount of fuel offloaded by tankers	1324.598
Amount of fuel used by RGs	1630.385

B.3 Best Solution Symmetric Group Representative

(0 18)(2 27 58 19 20 75 37 38 39)(3 40 41)(6 32 33 56 25 26)(7 30 31 55 44 45)(8 46 47)(9 23 24 59 48 49)(10 34 35)(11 21 22 57 42 43)(12 28 29 53 36) on 80 letters

B.4 Tanker Assignments

The following lists the information associated with the assignments of the active tankers in the best solution reported.

Tanker: 0

Type: KC135R

Beddown: KBGR

Take-off time: 8.483*

Arrive Node 18 at time: 9.393*

Node 18 is at location: AR12 for RG 0

Distance traveled: 391.169*

Fuel burned this leg: 17.917*

Fuel available for offload: 152.083*

Wait at Node 18 for time: 0.000*

Service Node 18 for time: 0.356*

Offloading fuel amt (in klbs.): 72.553*

Fuel remaining after offload: 79.530*

Depart Node 18 at time: 9.748*

Return home at time: 10.658*

Distance traveled: 391.169*

Total Distance traveled: 782.339*

Total fuel offloaded: 72.553*

Total fuel burned: 25.462*

Total fuel used: 98.015*

Tanker: 2

Type: KC135R

Beddown: KBGR

Take-off time: 0.483*

Arrive Node 27 at time: 1.393*

Node 27 is at location: AR12 for RG 1

Distance traveled: 391.169*

Fuel burned this leg: 17.917*

Fuel available for offload: 152.083*

Wait at Node 27 for time: 0.000*

Service Node 27 for time: 0.356*

Offloading fuel amt (in klbs.): 72.553*

Fuel remaining after offload: 79.530*

Depart Node 27 at time: 1.748*

Arrive Node 58 at time: 2.658*

Node 58 is at location: KBGR Distance traveled: 391.169*

Fuel burned this leg: 7.544*

Fuel available for offload: 71.985*

Wait at Node 58 for time: 2.181*

Service Node 58 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: KBGR

Depart Node 58 at time: 8.838*

Arrive Node 19 at time: 9.748*

Node 19 is at location: AR12 for RG 0

Distance traveled: 391.169*

Fuel burned this leg: 15.424*

Fuel available for offload: 154.576*

Wait at Node 19 for time: 0.000*

Service Node 19 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 154.576*

Depart Node 19 at time: 9.748*

Arrive Node 20 at time: 12.024*

Node 20 is at location: AR15 for RG 0

Distance traveled: 978.775*

Fuel burned this leg: 16.783*

Fuel available for offload: 137.793*

Wait at Node 20 for time: 0.000*

Service Node 20 for time: 0.368*

Offloading fuel amt (in klbs.): 75.073*

Fuel remaining after offload: 62.719*

Depart Node 20 at time: 12.392*

Arrive Node 75 at time: 15.517*

Node 75 is at location: KBGR Distance traveled: 1343.357*

Fuel burned this leg: 12.073*

Fuel available for offload: 50.647*

Wait at Node 75 for time: 0.000*

Service Node 75 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: KBGR

Depart Node 75 at time: 19.517*

Arrive Node 37 at time: 20.160*

Node 37 is at location: AR19 for RG 2

Distance traveled: 276.808*

Fuel burned this leg: 10.407*

Fuel available for offload: 159.593*

Wait at Node 37 for time: 0.000*

Service Node 37 for time: 0.397*

Offloading fuel amt (in klbs.): 54.723*

Fuel remaining after offload: 104.870*

Depart Node 37 at time: 20.557*

Arrive Node 38 at time: 20.557*

Node 38 is at location: AR19 for RG 2

Distance traveled: 0.000*

Fuel burned this leg: 0.000*

Fuel available for offload: 104.870*

Wait at Node 38 for time: 0.000*

Service Node 38 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 104.870*

Depart Node 38 at time: 20.557*

Arrive Node 39 at time: 23.130*

Node 39 is at location: AR18 for RG 2

Distance traveled: 1106.617*

Fuel burned this leg: 1.289*

Fuel available for offload: 103.581*

Wait at Node 39 for time: 0.000*

Service Node 39 for time: 0.647*

Offloading fuel amt (in klbs.): 89.281*

Fuel remaining after offload: 14.300*

Depart Node 39 at time: 23.777*

Return home at time: 25.756*

Distance traveled: 850.819*

Total Distance traveled: 5729.885*

Total fuel offloaded: 291.631*

Total fuel burned: 73.200*

Total fuel used: 364.830*

Tanker: 3

Type: KC135R

Beddown: KBGR

Take-off time: 21.799*

Arrive Node 40 at time: 23.777*

Node 40 is at location: AR18 for RG 2

Distance traveled: 850.819*

Fuel burned this leg: 29.187*

Fuel available for offload: 140.813*

Wait at Node 40 for time: 0.000*

Service Node 40 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 140.813*

Depart Node 40 at time: 23.777*

Arrive Node 41 at time: 25.504*

Node 41 is at location: AR20 for RG 2

Distance traveled: 742.609*

Fuel burned this leg: 17.222*

Fuel available for offload: 123.590*

Wait at Node 41 for time: 0.000*

Service Node 41 for time: 0.449*

Offloading fuel amt (in klbs.): 61.971*

Fuel remaining after offload: 61.619*

Depart Node 41 at time: 25.953*

Return home at time: 29.657*

Distance traveled: 1592.747*

Total Distance traveled: 3186.175*

Total fuel offloaded: 61.971*

Total fuel burned: 73.666*

Total fuel used: 135.636*

Tanker: 6

Type: KC135R

Beddown: EGUN

Take-off time: 4.497*

Arrive Node 32 at time: 7.108*

Node 32 is at location: AR14 for RG 1

Distance traveled: 1122.544*

Fuel burned this leg: 35.785*

Fuel available for offload: 134.215*

Wait at Node 32 for time: 0.000*

Service Node 32 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 134.215*

Depart Node 32 at time: 7.108*

Arrive Node 33 at time: 9.364*

Node 33 is at location: AR16 for RG 1

Distance traveled: 969.893*

Fuel burned this leg: 21.942*

Fuel available for offload: 112.273*

Wait at Node 33 for time: 0.000*

Service Node 33 for time: 0.365*

Offloading fuel amt (in klbs.): 74.425*

Fuel remaining after offload: 37.848*

Depart Node 33 at time: 9.728*

Arrive Node 56 at time: 11.679*

Node 56 is at location: EGUN Distance traveled: 838.604*

Fuel burned this leg: 13.246*

Fuel available for offload: 24.602*

Wait at Node 56 for time: 0.100*

Service Node 56 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: EGUN

Depart Node 56 at time: 15.778*

Arrive Node 25 at time: 17.728*

Node 25 is at location: AR16 for RG 0

Distance traveled: 838.604*

Fuel burned this leg: 20.417*

Fuel available for offload: 149.583*

Wait at Node 25 for time: 0.000*

Service Node 25 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 149.583*

Depart Node 25 at time: 17.728*

Arrive Node 26 at time: 20.048*

Node 26 is at location: AR13 for RG 0

Distance traveled: 997.421*

Fuel burned this leg: 12.726*

Fuel available for offload: 136.857*

Wait at Node 26 for time: 0.000*

Service Node 26 for time: 0.374*

Offloading fuel amt (in klbs.): 76.370*

Fuel remaining after offload: 60.487*

Depart Node 26 at time: 20.422*

Return home at time: 23.430*

Distance traveled: 1293.264*

Total Distance traveled: 6060.330*

Total fuel offloaded: 150.795*

Total fuel burned: 109.390*

Total fuel used: 260.185*

Tanker: 7

Type: KC135R

Beddown: EGUN

Take-off time: 0.118*

Arrive Node 30 at time: 4.392*

Node 30 is at location: AR15 for RG 1

Distance traveled: 1837.982*

Fuel burned this leg: 52.391*

Fuel available for offload: 117.609*

Wait at Node 30 for time: 0.000*

Service Node 30 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 117.609*

Depart Node 30 at time: 4.392*

Arrive Node 31 at time: 6.730*

Node 31 is at location: AR14 for RG 1

Distance traveled: 1005.373*

Fuel burned this leg: 21.622*

Fuel available for offload: 95.987*

Wait at Node 31 for time: 0.000*

Service Node 31 for time: 0.378*

Offloading fuel amt (in klbs.): 77.017*

Fuel remaining after offload: 18.969*

Depart Node 31 at time: 7.108*

Arrive Node 55 at time: 9.719*

Node 55 is at location: EGUN Distance traveled: 1122.544*

Fuel burned this leg: 15.726*

Fuel available for offload: 3.243*

Wait at Node 55 for time: 11.845*

Service Node 55 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: EGUN

Depart Node 55 at time: 25.563*

Arrive Node 44 at time: 28.174*

Node 44 is at location: AR14 for RG 2

Distance traveled: 1122.544*

Fuel burned this leg: 22.294*

Fuel available for offload: 147.706*

Wait at Node 44 for time: 0.000*

Service Node 44 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 147.706*

Depart Node 44 at time: 28.174*

Arrive Node 45 at time: 30.429*

Node 45 is at location: AR16 for RG 2

Distance traveled: 969.893*

Fuel burned this leg: 10.369*

Fuel available for offload: 137.337*

Wait at Node 45 for time: 0.000*

Service Node 45 for time: 0.574*

Offloading fuel amt (in klbs.): 79.184*

Fuel remaining after offload: 58.154*

Depart Node 45 at time: 31.003*

Return home at time: 32.953*

Distance traveled: 838.604*

Total Distance traveled: 6896.939*

Total fuel offloaded: 156.201*

Total fuel burned: 123.760*

Total fuel used: 279.961*

Tanker: 8

Type: KC135R

Beddown: EGUN

Take-off time: 29.053*

Arrive Node 46 at time: 31.003*

Node 46 is at location: AR16 for RG 2

Distance traveled: 838.604*

Fuel burned this leg: 28.979*

Fuel available for offload: 141.021*

Wait at Node 46 for time: 0.000*

Service Node 46 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 141.021*

Depart Node 46 at time: 31.003*

Arrive Node 47 at time: 33.323*

Node 47 is at location: AR13 for RG 2

Distance traveled: 997.421*

Fuel burned this leg: 22.925*

Fuel available for offload: 118.096*

Wait at Node 47 for time: 0.000*

Service Node 47 for time: 0.591*

Offloading fuel amt (in klbs.): 81.493*

Fuel remaining after offload: 36.603*

Depart Node 47 at time: 33.913*

Return home at time: 36.921*

Distance traveled: 1293.264*

Total Distance traveled: 3129.290*

Total fuel offloaded: 81.493*

Total fuel burned: 71.731*

Total fuel used: 153.224*

Tanker: 9

Type: KC135R

Beddown: EGUN

Take-off time: 12.497*

Arrive Node 23 at time: 15.108*

Node 23 is at location: AR14 for RG 0

Distance traveled: 1122.544*

Fuel burned this leg: 35.785*

Fuel available for offload: 134.215*

Wait at Node 23 for time: 0.000*

Service Node 23 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 134.215*

Depart Node 23 at time: 15.108*

Arrive Node 24 at time: 17.364*

Node 24 is at location: AR16 for RG 0

Distance traveled: 969.893*

Fuel burned this leg: 21.942*

Fuel available for offload: 112.273*

Wait at Node 24 for time: 0.000*

Service Node 24 for time: 0.365*

Offloading fuel amt (in klbs.): 74.425*

Fuel remaining after offload: 37.848*

Depart Node 24 at time: 17.728*

Arrive Node 59 at time: 19.679*

Node 59 is at location: EGUN Distance traveled: 838.604*

Fuel burned this leg: 13.246*

Fuel available for offload: 24.602*

Wait at Node 59 for time: 7.227*

Service Node 59 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: EGUN

Depart Node 59 at time: 30.906*

Arrive Node 48 at time: 33.913*

Node 48 is at location: AR13 for RG 2

Distance traveled: 1293.264*

Fuel burned this leg: 26.647*

Fuel available for offload: 143.353*

Wait at Node 48 for time: 0.000*

Service Node 48 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 143.353*

Depart Node 48 at time: 33.913*

Arrive Node 49 at time: 35.666*

Node 49 is at location: AR17 for RG 2

Distance traveled: 753.619*

Fuel burned this leg: 9.339*

Fuel available for offload: 134.014*

Wait at Node 49 for time: 0.000*

Service Node 49 for time: 0.457*

Offloading fuel amt (in klbs.): 62.999*

Fuel remaining after offload: 71.015*

Depart Node 49 at time: 36.122*

Return home at time: 40.720*

Distance traveled: 1977.062*

Total Distance traveled: 6954.986*

Total fuel offloaded: 137.424*

Total fuel burned: 116.793*

Total fuel used: 254.217*

Tanker: 10

Type: KC135R

Beddown: EGUN

Take-off time: 7.778*

Arrive Node 34 at time: 9.728*

Node 34 is at location: AR16 for RG 1

Distance traveled: 838.604*

Fuel burned this leg: 28.979*

Fuel available for offload: 141.021*

Wait at Node 34 for time: 0.000*

Service Node 34 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 141.021*

Depart Node 34 at time: 9.728*

Arrive Node 35 at time: 12.048*

Node 35 is at location: AR13 for RG 1

Distance traveled: 997.421*

Fuel burned this leg: 22.925*

Fuel available for offload: 118.096*

Wait at Node 35 for time: 0.000*

Service Node 35 for time: 0.374*

Offloading fuel amt (in klbs.): 76.370*

Fuel remaining after offload: 41.726*

Depart Node 35 at time: 12.422*

Return home at time: 15.430*

Distance traveled: 1293.264*

Total Distance traveled: 3129.290*

Total fuel offloaded: 76.370*

Total fuel burned: 72.267*

Total fuel used: 148.636*

Tanker: 11

Type: KC135R

Beddown: EGUN

Take-off time: 8.118*

Arrive Node 21 at time: 12.392*

Node 21 is at location: AR15 for RG 0

Distance traveled: 1837.982*

Fuel burned this leg: 52.391*

Fuel available for offload: 117.609*

Wait at Node 21 for time: 0.000*

Service Node 21 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 117.609*

Depart Node 21 at time: 12.392*

Arrive Node 22 at time: 14.730*

Node 22 is at location: AR14 for RG 0

Distance traveled: 1005.373*

Fuel burned this leg: 21.622*

Fuel available for offload: 95.987*

Wait at Node 22 for time: 0.000*

Service Node 22 for time: 0.378*

Offloading fuel amt (in klbs.): 77.017*

Fuel remaining after offload: 18.969*

Depart Node 22 at time: 15.108*

Arrive Node 57 at time: 17.719*

Node 57 is at location: EGUN Distance traveled: 1122.544*

Fuel burned this leg: 15.726*

Fuel available for offload: 3.243*

Wait at Node 57 for time: 0.398*

Service Node 57 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: EGUN

Depart Node 57 at time: 22.117*

Arrive Node 42 at time: 25.953*

Node 42 is at location: AR20 for RG 2

Distance traveled: 1649.668*

Fuel burned this leg: 28.410*

Fuel available for offload: 141.590*

Wait at Node 42 for time: 0.000*

Service Node 42 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 141.590*

Depart Node 42 at time: 25.953*

Arrive Node 43 at time: 27.715*

Node 43 is at location: AR14 for RG 2

Distance traveled: 757.414*

Fuel burned this leg: 7.765*

Fuel available for offload: 133.825*

Wait at Node 43 for time: 0.000*

Service Node 43 for time: 0.459*

Offloading fuel amt (in klbs.): 63.342*

Fuel remaining after offload: 70.483*

Depart Node 43 at time: 28.174*

Return home at time: 30.784*

Distance traveled: 1122.544*

Total Distance traveled: 7495.525*

Total fuel offloaded: 140.359*

Total fuel burned: 129.197*

Total fuel used: 269.556*

Tanker: 12

Type: KC135R

Beddown: KGSB

Take-off time: 0.010*

Arrive Node 28 at time: 1.748*

Node 28 is at location: AR12 for RG 1

Distance traveled: 747.410*

Fuel burned this leg: 26.685*

Fuel available for offload: 143.315*

Wait at Node 28 for time: 0.000*

Service Node 28 for time: 0.000*

Offloading fuel amt (in klbs.): 0.000*

Fuel remaining after offload: 143.315*

Depart Node 28 at time: 1.748*

Arrive Node 29 at time: 4.024*

Node 29 is at location: AR15 for RG 1

Distance traveled: 978.775*

Fuel burned this leg: 22.679*

Fuel available for offload: 120.636*

Wait at Node 29 for time: 0.000*

Service Node 29 for time: 0.368*

Offloading fuel amt (in klbs.): 75.073*

Fuel remaining after offload: 45.562*

Depart Node 29 at time: 4.392*

Arrive Node 53 at time: 7.517*

Node 53 is at location: KBGR Distance traveled: 1343.357*

Fuel burned this leg: 21.541*

Fuel available for offload: 24.022*

Wait at Node 53 for time: 4.465*

Service Node 53 for time: 4.000*

Tanker refueled to (in klbs.): 170.000*

At Base: KBGR

Depart Node 53 at time: 15.981*

Arrive Node 36 at time: 18.118*

Node 36 is at location: KSZL for RG 2

Distance traveled: 918.607*

Fuel burned this leg: 21.458*

Fuel available for offload: 148.542*

Wait at Node 36 for time: 0.000*

Service Node 36 for time: 0.585*

Offloading fuel amt (in klbs.): 80.730*

Fuel remaining after offload: 67.812*

Depart Node 36 at time: 18.703*

Return home at time: 20.501*

Distance traveled: 773.445*

Total Distance traveled: 4761.594*

Total fuel offloaded: 155.803*

Total fuel burned: 96.159*

Total fuel used: 251.963*

B.5 Receiver Group Details

The following lists the information associated with the assignments of the RGs in the best solution reported.

RG: 0 with 6 F15

Depart from location KLFI at time: 8.000*

Arrive Location AR12 at time: 9.393*

This is node 18 serviced by tanker 0

This is node 19 serviced by tanker 2

Distance traveled: 618.285*

Wait at AR12 for time: 0.000*

Service at AR12 for time: 0.356*

Depart Location AR12 at time: 9.748*

Arrive Location AR15 at time: 12.024*

This is node 20 serviced by tanker 2

This is node 21 serviced by tanker 11

Distance traveled: 978.775*

Wait at AR15 for time: 0.000*

Service at AR15 for time: 0.368*

Depart Location AR15 at time: 12.392*

Arrive Location AR14 at time: 14.730*

This is node 22 serviced by tanker 11

This is node 23 serviced by tanker 9

Distance traveled: 1005.373*

Wait at AR14 for time: 0.000*

Service at AR14 for time: 0.378*

Depart Location AR14 at time: 15.108*

Arrive Location AR16 at time: 17.364*

This is node 24 serviced by tanker 9

This is node 25 serviced by tanker 6

Distance traveled: 969.893*

Wait at AR16 for time: 0.000*

Service at AR16 for time: 0.365*

Depart Location AR16 at time: 17.728*

Arrive Location AR13 at time: 20.048*

This is node 26 serviced by tanker 6

Distance traveled: 997.421*

Wait at AR13 for time: 0.000*

Service at AR13 for time: 0.374*

Depart Location AR13 at time: 20.422*

Arrive Location OERY at time: 23.891*

Distance traveled: 1540.256*

Total Distance traveled: 6110.003*

Total Fuel Uploaded: 375.438*

Total Fuel Used: 499.469*

Total fuel Available: 500.820*

RG: 1 with 6 F15

Depart from location KLFI at time: 0.000*

Arrive Location AR12 at time: 1.393*

This is node 27 serviced by tanker 2

This is node 28 serviced by tanker 12

Distance traveled: 618.285*

Wait at AR12 for time: 0.000*

Service at AR12 for time: 0.356*

Depart Location AR12 at time: 1.748*

Arrive Location AR15 at time: 4.024*

This is node 29 serviced by tanker 12

This is node 30 serviced by tanker 7

Distance traveled: 978.775*

Wait at AR15 for time: 0.000*

Service at AR15 for time: 0.368*

Depart Location AR15 at time: 4.392*

Arrive Location AR14 at time: 6.730*

This is node 31 serviced by tanker 7

This is node 32 serviced by tanker 6

Distance traveled: 1005.373*

Wait at AR14 for time: 0.000*

Service at AR14 for time: 0.378*

Depart Location AR14 at time: 7.108*

Arrive Location AR16 at time: 9.364*

This is node 33 serviced by tanker 6

This is node 34 serviced by tanker 10

Distance traveled: 969.893*

Wait at AR16 for time: 0.000*

Service at AR16 for time: 0.365*

Depart Location AR16 at time: 9.728*

Arrive Location AR13 at time: 12.048*

This is node 35 serviced by tanker 10

Distance traveled: 997.421*

Wait at AR13 for time: 0.000*

Service at AR13 for time: 0.374*

Depart Location AR13 at time: 12.422*

Arrive Location OERY at time: 15.891*

Distance traveled: 1540.256*

Total Distance traveled: 6110.003*

Total Fuel Uploaded: 375.438*

Total Fuel Used: 499.469*

Total fuel Available: 500.820*

RG: 2 with 6 F117

Depart from location KHMN at time: 16.535*

Arrive Location KSZL at time: 18.118*

This is node 36 serviced by tanker 12

Distance traveled: 704.505*

Wait at KSZL for time: 0.000*

Service at KSZL for time: 0.585*

Depart Location KSZL at time: 18.703*

Arrive Location AR19 at time: 20.160*

This is node 37 serviced by tanker 2

This is node 38 serviced by tanker 2

Distance traveled: 648.617*

Wait at AR19 for time: 0.000*

Service at AR19 for time: 0.397*

Depart Location AR19 at time: 20.557*

Arrive Location AR18 at time: 23.130*

This is node 39 serviced by tanker 2

This is node 40 serviced by tanker 3

Distance traveled: 1106.617*

Wait at AR18 for time: 0.000*

Service at AR18 for time: 0.647*

Depart Location AR18 at time: 23.777*

Arrive Location AR20 at time: 25.504*

This is node 41 serviced by tanker 3

This is node 42 serviced by tanker 11

Distance traveled: 742.609*

Wait at AR20 for time: 0.000*

Service at AR20 for time: 0.449*

Depart Location AR20 at time: 25.953*

Arrive Location AR14 at time: 27.715*

This is node 43 serviced by tanker 11

This is node 44 serviced by tanker 7

Distance traveled: 757.414*

Wait at AR14 for time: 0.000*

Service at AR14 for time: 0.459*

Depart Location AR14 at time: 28.174*

Arrive Location AR16 at time: 30.429*

This is node 45 serviced by tanker 7

This is node 46 serviced by tanker 8

Distance traveled: 969.893*

Wait at AR16 for time: 0.000*

Service at AR16 for time: 0.574*

Depart Location AR16 at time: 31.003*

Arrive Location AR13 at time: 33.323*

This is node 47 serviced by tanker 8

This is node 48 serviced by tanker 9

Distance traveled: 997.421*

Wait at AR13 for time: 0.000*

Service at AR13 for time: 0.591*

Depart Location AR13 at time: 33.913*

Arrive Location AR17 at time: 35.666*

This is node 49 serviced by tanker 9

Distance traveled: 753.619*

Wait at AR17 for time: 0.000*

Service at AR17 for time: 0.457*

Depart Location AR17 at time: 36.122*

Arrive Location OEDR at time: 38.194*

Distance traveled: 921.732*

Total Distance traveled: 7602.427*

Total Fuel Uploaded: 573.722*

Total Fuel Used: 663.529*

Total fuel Available: 666.722*

B.6 Node Details

The following lists the information associated with the assignments of the Nodes in the best solution reported.

Node 0 at location KBGR

Tanker 0 started at 8.483*

Tanker 0 finished at 10.658*

Node 2 at location KBGR

Tanker 2 started at 0.483*

Tanker 2 finished at 25.756*

Node 3 at location KBGR

Tanker 3 started at 21.799*

Tanker 3 finished at 29.657*

Node 6 at location EGUN

Tanker 6 started at 4.497*

Tanker 6 finished at 23.430*

Node 7 at location EGUN

Tanker 7 started at 0.118*

Tanker 7 finished at 32.953*

Node 8 at location EGUN

Tanker 8 started at 29.053*

Tanker 8 finished at 36.921*

Node 9 at location EGUN

Tanker 9 started at 12.497*

Tanker 9 finished at 40.720*

Node 10 at location EGUN

Tanker 10 started at 7.778*

Tanker 10 finished at 15.430*

Node 11 at location EGUN

Tanker 11 started at 8.118*

Tanker 11 finished at 30.784*

Node 12 at location KGSB

Tanker 12 started at 0.010*

Tanker 12 finished at 20.501*

Node 18 at location AR12

Tanker 0 arrival at 9.393*

RG 0 arrival at 9.393*

Service Finished at 9.748*

Node 19 at location AR12

Tanker 2 arrival at 9.748*

RG 0 arrival at 9.748*

Service Finished at 9.748*

Node 20 at location AR15

Tanker 2 arrival at 12.024*

RG 0 arrival at 12.024*

Service Finished at 12.392*

Node 21 at location AR15

Tanker 11 arrival at 12.392*

RG 0 arrival at 12.392*

Service Finished at 12.392*

Node 22 at location AR14

Tanker 11 arrival at 14.730*

RG 0 arrival at 14.730*

Service Finished at 15.108*

Node 23 at location AR14

Tanker 9 arrival at 15.108*

RG 0 arrival at 15.108*

Service Finished at 15.108*

Node 24 at location AR16

Tanker 9 arrival at 17.364*

RG 0 arrival at 17.364*

Service Finished at 17.728*

Node 25 at location AR16

Tanker 6 arrival at 17.728*

RG 0 arrival at 17.728*

Service Finished at 17.728*

Node 26 at location AR13

Tanker 6 arrival at 20.048*

RG 0 arrival at 20.048*

Service Finished at 20.422*

Node 27 at location AR12

Tanker 2 arrival at 1.393*

RG 1 arrival at 1.393*

Service Finished at 1.748*

Node 28 at location AR12

Tanker 12 arrival at 1.748*

RG 1 arrival at 1.748*

Service Finished at 1.748*

Node 29 at location AR15

Tanker 12 arrival at 4.024*

RG 1 arrival at 4.024*

Service Finished at 4.392*

Node 30 at location AR15

Tanker 7 arrival at 4.392*

RG 1 arrival at 4.392*

Service Finished at 4.392*

Node 31 at location AR14

Tanker 7 arrival at 6.730*

RG 1 arrival at 6.730*

Service Finished at 7.108*

Node 32 at location AR14

Tanker 6 arrival at 7.108*

RG 1 arrival at 7.108*

Service Finished at 7.108*

Node 33 at location AR16

Tanker 6 arrival at 9.364*

RG 1 arrival at 9.364*

Service Finished at 9.728*

Node 34 at location AR16

Tanker 10 arrival at 9.728*

RG 1 arrival at 9.728*

Service Finished at 9.728*

Node 35 at location AR13

Tanker 10 arrival at 12.048*

RG 1 arrival at 12.048*

Service Finished at 12.422*

Node 36 at location KSZL

Tanker 12 arrival at 18.118*

RG 2 arrival at 18.118*

Service Finished at 18.703*

Node 37 at location AR19

Tanker 2 arrival at 20.160*

RG 2 arrival at 20.160*

Service Finished at 20.557*

Node 38 at location AR19

Tanker 2 arrival at 20.557*

RG 2 arrival at 20.557*

Service Finished at 20.557*

Node 39 at location AR18

Tanker 2 arrival at 23.130*

RG 2 arrival at 23.130*

Service Finished at 23.777*

Node 40 at location AR18

Tanker 3 arrival at 23.777*

RG 2 arrival at 23.777*

Service Finished at 23.777*

Node 41 at location AR20

Tanker 3 arrival at 25.504*

RG 2 arrival at 25.504*

Service Finished at 25.953*

Node 42 at location AR20

Tanker 11 arrival at 25.953*

RG 2 arrival at 25.953*

Service Finished at 25.953*

Node 43 at location AR14

Tanker 11 arrival at 27.715*

RG 2 arrival at 27.715*

Service Finished at 28.174*

Node 44 at location AR14

Tanker 7 arrival at 28.174*

RG 2 arrival at 28.174*

Service Finished at 28.174*

Node 45 at location AR16

Tanker 7 arrival at 30.429*

RG 2 arrival at 30.429*

Service Finished at 31.003*

Node 46 at location AR16

Tanker 8 arrival at 31.003*

RG 2 arrival at 31.003*

Service Finished at 31.003*

Node 47 at location AR13

Tanker 8 arrival at 33.323*

RG 2 arrival at 33.323*

Service Finished at 33.913*

Node 48 at location AR13

Tanker 9 arrival at 33.913*

RG 2 arrival at 33.913*

Service Finished at 33.913*

Node 49 at location AR17

Tanker 9 arrival at 35.666*

RG 2 arrival at 35.666*

Service Finished at 36.122*

Node 50 at location PAEI

Not in current solution

Node 51 at location KGSB

Not in current solution

Node 52 at location EGUN

Not in current solution

Node 53 at location KBGR

Tanker 12 arrival at 7.517*

for later reuse.

Service Finished at 11.517*

Node 54 at location EGUN

Not in current solution

Node 55 at location EGUN

Tanker 7 arrival at 9.719*

for later reuse.

Service Finished at 13.719*

Node 56 at location EGUN

Tanker 6 arrival at 11.679*

for later reuse.

Service Finished at 15.679*

Node 57 at location EGUN

Tanker 11 arrival at 17.719*

for later reuse.

Service Finished at 21.719*

Node 58 at location KBGR

Tanker 2 arrival at 2.658*

for later reuse.

Service Finished at 6.658*

Node 59 at location EGUN

Tanker 9 arrival at 19.679*

for later reuse.

Service Finished at 23.679*

Node 60 at location EGUN

Not in current solution

Node 61 at location EGUN

Not in current solution

Node 62 at location KBGR

Not in current solution

Node 63 at location KBGR

Not in current solution

Node 64 at location KBGR

Not in current solution

Node 65 at location KGSB

Not in current solution

Node 66 at location KBGR

Not in current solution

Node 67 at location KBGR

Not in current solution

Node 68 at location KBGR

Not in current solution

Node 69 at location KGSB

Not in current solution

Node 70 at location KGSB

Not in current solution

Node 71 at location KGSB

Not in current solution

Node 72 at location KGSB

Not in current solution

Node 73 at location KGSB

Not in current solution

Node 74 at location KBGR

Not in current solution

Node 75 at location KBGR

Tanker 2 arrival at 15.517*

for later reuse.

Service Finished at 19.517*

Node 76 at location KGSB

Not in current solution

Node 77 at location KGSB

Not in current solution

Node 78 at location KGSB

Not in current solution

Node 79 at location PAEI

Not in current solution

Appendix C

A Group Class Java[TM] Primer

A user guide intended to be a step-by-step instruction of how to create a group using the Java based Group class definition has been written by the author. This guide is available at

http://www.me.utexas.edu/~orie/GrpUserArtStyle.pdf

and demonstrates each step of the Group class implementation process using the Symmetric Group on n letters (S_n) as an illustration. The methods defined for S_n within this guide are meant to be the minimal set of necessary methods to implement any group using the Group class interface. The implemented methods are not intended to restrict the user from creating other useful methods. For examples of additional methods that have been successfully applied to S_n, see the SymmetricGroup User's guide located at

http://www.me.utexas.edu/~orie/SymGroupArtStyle.pdf

A brief description of this guide is given in Appendix D.

Appendix D

A SymmetricGroup Class Java^{TM} Primer

The Symmetric Group on n-letters (S_n) is an implementation of the Group class mentioned in Appendix C. The details of this implementation are provided in a user guide at

http://www.me.utexas.edu/~orie/SymGroupArtStyle.pdf

as step-by-step instructions along with examples of Java^{TM}based code.

S_n, by its inherent structure, easily represents solutions to Partitioning and Ordering Problems (P|O). The cyclic structure of S_n captures the partitioning aspect while the arrangement of letters within each cycle determines the ordering aspect. This manual is intended to provide details of how the SymmetricGroup class has been derived and how it can be used for any P|O. Through the use of the group actions of conjugation and multiplication, any solution to a P|O is reachable from another solution.

The SymmetricGroup class guide provides a description of each public method and provides some examples of how to apply the S_n within the Tabu Search framework established by Harder (2000). The Java^{TM} archive file (*.jar) that contains the Group and SymmetricGroup classes is available for download at

http://www.me.utexas.edu/~orie/techrep.html

under Technical Report ORP00-04.

Index

Bibliography

Aarts, E., Van Laarhooven, P., Lenstra, J. & Ulder, N. (1994). A Computational Study of Local Search Algorithms for Job-Shop Scheduling, *ORSA Journal on Computing* **6**(2): 118–125.

Aboudi, R. & Jornsten, K. (1994). Tabu Search for General Zero-One Integer Programs Using the Pivot and Complement Heuristic, *ORSA Journal on Computing* **6**(1): 82–93.

Adams, J., Balas, E. & Zawack, D. (1988). The Shifting Bottleneck Procedure for Job-Shop Scheduling, *Management Science* **34**(3): 391–401.

AFDD 2-6.2 (1999). Air Refueling: Air Force Doctrine Document 2-6.2, Part of Joint Publication 3-17, Joint Tactics, Techniques, and Procedures for Air Mobility Operations.

Applegate, D. & Cook, W. (1991). A Computational Study of the Job-Shop Scheduling Problem, *ORSA Journal on Computing* **3**(2): 149–156.

Artin, M. (1991). *Algebra*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.

Assad, A. A. (1988). *Modeling and Implementation Issues in Vehicle Routing*, North-Holland, chapter 1, pp. 7–45.

Atkinson, J. B. (1998). A Greedy Randomised Search Heuristic for Time-Constrained Vehicle Scheduling and the Incorporation of a Learning Strategy, *Journal of the Operational Research Society* **49**(7): 700–708.

Augerat, P., Belenguer, J. M., Benavent, E., Corberan, A. & Naddef, D. (1998). Separating Capacity Constraints in the CVRP Using Tabu Search, *European Journal of Operational Research* **106**: 546–557.

Averbakh, I. & Berman, O. (1999). A Simple Heuristic for M-Machine Flow-Shop and Its Application in Routing-Scheduling Problems, *Operations Research* **47**(1): 165–170.

Baker, B. M. & Sheasby, J. (1999a). Accelerating the Convergence of Subgradient Optimisation, *European Journal of Operational Research* **117**: 136–144.

Baker, B. M. & Sheasby, J. (1999b). Extensions to the Generalised Assignment Heuristic for Vehicle Routing, *European Journal of Operational Research* **119**: 147–157.

Baker, E. K. & Schaffer, J. R. (1986). Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem, *American Journal of Mathematical and Management Sciences* **16**: 261–300.

Baker, K. (1974). *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York.

Balas, E. (1969). Machine Scheduling Via Disjunctive Graphs: An Implicit Enumeration Algorithm, *Operations Research* **17**: 941–957.

Balas, E. (1983). A Class of Location, Distribution and Scheduling Problems: Modeling and Solution Methods, *in* P. Gray & L. Yuanzhang (eds), *Proceedings of the Chinese-U.S. Symposium on Systems Analysis*, John Wiley & Sons.

Balas, E. & Carrera, M. (1996). A Dynamic Subgradient-Based Branch and Bound Procedure for Set Covering, *Operations Research* **44**: 875–890.

Balas, E. & Ho, A. (1980). Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization: A Computational Study, *Mathematical Programming Study* **12**: 37–60.

Barner, K. E. (1997). Design of Permutation Order Statistic Filters Through Group Colorings, *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing* **44**(7): 531–548.

Barnes, J. & Chambers, J. (1995). Solving the Job Shop Scheduling Problem Using Tabu Search, *IIE Transactions* **27**: 257–263.

Barnes, J. W. & Acevedo, R. (2000). A Simple Relation Between Multiplicative Moves and Conjugation, *Technical Report Series ORP00-03*, The University of Texas at Austin, Graduate Program in Operations Research.
(`http://www.me.utexas.edu/~orie/techrep.html`).

Barnes, J. W. & Carlton, W. B. (1995). Solving the Vehicle Routing Problem with Time Windows Using Reactive Tabu Search.

Barnes, J. W. & Carlton, W. B. (1996). A Hierarchical Classification Scheme for the General Vehicle Routing Problem.

Barnes, J. W. & Colletti, B. W. (1999). Group Theory and Metaheuristic Search Neighborhoods, *Technical Report Series ORP99-02*, The University of Texas at Austin. Submitted to *Journal of Heuristics.*

Barnes, J. W., Colletti, B. W. & Neuway, D. L. (2000a). Group Theory and Transition Matrices for Conjugative Move Disciplines in Multiple TSP's with Known Solution Structure, *Technical Report Series ORP00-02*, The University of Texas at Austin, Graduate Program in Operations Research.
(`http://www.me.utexas.edu/~orie/techrep.html`).

Barnes, J. W., Colletti, B. W. & Neuway, D. L. (2000b). Using Group Theory and Transition Matrices to Study a Class of Metaheuristic Neighborhoods, *Technical Report Series ORP00-01*, The University of Texas at Austin, Graduate Program in Operations Research. submitted to *European Journal of Operational Research*
(`http://www.me.utexas.edu/~orie/techrep.html`).

Barnes, J. W., Wiley, V., Neuway, D., Colletti, B. & Solomon, A. (2001). Making the n-Cycles Within S_n Into a Group (Expanded Version), *Technical Report Series ORP01-01*, The University of Texas at Austin, Graduate Program in Operations Research.
(`http://www.me.utexas.edu/~orie/techrep.html`).

Bartalos, I., Dudas, T. & Imreh, B. (1995). On a Tour Construction Heuristic for the Asymmetric TSP, *Acta Cybernetica* **12**: 209–216.

Battiti, R. & Bertossi, A. A. (1999). Greedy, Prohibition, and Reactive Heuristics for Graph Partitioning, *IEEE Transactions on Computers* **48**(4): 361–385.

Battiti, R. & Tecchiolli, G. (1994). The Reactive Tabu Search, *ORSA Journal on Computing* **6**(2): 126–140.

Beasley, J. & Chu, P. (1996). A Genetic Algorithm for the Set Covering Problem, *European Journal of Operational Research* **94**: 392–404.

Beasley, J. & Jörnsten, K. (1992). Enhancing an Algorithm for Set Covering Problems, *European Journal of Operational Research* **58**: 293–300.

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G. & Prutzman, P. J. (1983). Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer, *Interfaces* **13**: 4–23.

Bertsimas, D. & Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*, Athena Scientific, Belmont, Massachusetts.

Blue, J. A. & Bennett, K. P. (1998). Hybrid Extreme Point Tabu Search, *European Journal of Operational Research* **106**: 676–688.

Boardman, B. S., Malstrom, E. M., Butler, D. P. & Cole, M. H. (1997). Computer Assisted Routing of Intermodal Shipments, *Computers & Industrial Engineering* **33**(1,2): 311–314.

Bourjolly, J.-M., Tomiuk, D. & Kapantow, G. H. M. (1999). Using Simulated Annealing to Minimize the Cost of Centralized Telecommunications Networks, *Infor* **37**(3): 271–296.

Brusco, M. J. & Jacobs, L. W. (1998). Personnel Tour Scheduling When Starting-Time Restrictions Are Present, *Management Science* **44**(4): 534–547.

Brusco, M., Jacobs, L. & Thompson, G. (1996). A Morphing Procedure to Supplement a Simulated Annealing Heuristic for Cost- and Coverage-Correlated Weighted Set-Covering Problems, *Technical Report Working Paper*, Operations Management and Information Systems Department, Northern Illinois University.

Byrd, V. B. (1994). *Passing Gas: The History of Inflight Refueling*, Byrd Publishing Company, Chico, CA.

Calhoun, K. M. (2000). *A Tabu Search for Scheduling and Rescheduling Combat Aircraft*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Capehart, S. R. (2000). *A Tabu Search Metaheuristic for the Air Refueling Tanker Assignment Problem*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Caprara, A., Fischetti, M. & Toth, P. (1999). A Heuristic Method for the Set Covering Problem, *Operations Research* **47**(5): 730–743.

Caprara, A., Fischetti, M. & Toth, P. (2001). Algorithm for the Set Covering Problem. to appear in Annals of Operations Research.

Carlier, J. & Pinson, E. (1989). An Algorithm for Solving the Job Shop Problem, *Management Science* **35**(2): 164–176.

Carlton, W. B. (1995). *A Tabu Search to the General Vehicle Routing Problem*, PhD thesis, University of Texas, Austin, TX.

Carlton, W. B. & Barnes, J. W. (1996a). A Note on Hashing Functions and Tabu Search Algorithms, *European Journal of Operational Research* **106**: 237–239.

Carlton, W. B. & Barnes, J. W. (1996b). Solving the Traveling-Salesman Problem with Time Windows Using Tabu Search, *IIE Transactions* **28**(8): 617–629.

Ceria, S., Nobili, P. & Sassano, A. (1995). A Lagrangian-Based Heuristic for Large-Scale Set Covering Problems, *Technical Report R.406*, IASI-CNR, Rome, Italy. to appear in Mathematical Programming.

Chambers, J. B. (1996). *Classical and Flexible Job Shop Scheduling by Tabu Search*, PhD thesis, The University of Texas at Austin.

Chambers, J. B. & Barnes, J. W. (1996). New Tabu Search Results for the Job Shop Scheduling Problem, *Technical Report Series ORP96-06*, The University of Texas at Austin.

Chen, X., Wan, W. & Xu, X. (1998). Modeling Rolling Batch Planning as Vehicle Routing Problem with Time Windows, *Computers & Operations Research* **25**(12): 1127–1136.

Cheng, R., Gen, M. & Tsujimura, Y. (1996). A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms-I Representation, *Computers & Industrial Engineering* **30**(4): 983–997.

Chiang, W. C. & Russell, R. (1997). A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows, *ORSA Journal on Computing* **9**: 417–433.

Chocolaad, C. A. (1998). *Solving Geometric Knapsack Problems Using Tabu Search Heuristics*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Christofides, N. (1985). *Vehicle Routing*, John Wiley & Sons, Chichester, chapter 10, pp. 148–163.

Chung, Y. & Fischer, G. W. (1995). A Neural Algorithm for Finding the Shortest Flow Path for an Automated Guided Vehicle System, *IIE Transactions* **27**: 773–783.

Colletti, B., Barnes, J. & Dokov, S. (1999). A Note on Characterizing the K-OPT Neighborhood Via Group Theory, *Journal of Heuristics* **5**: 51–55.

Colletti, B. & Barnes, J. W. (1999a). Computing M-TSP Tourlength Differences Via Group Theory, *Technical Report Series ORP99-01*, The University of Texas at Austin, Graduate Program in Operations Research.

Colletti, B. & Barnes, J. W. (1999b). Using Group Theory to Efficiently Compute Move Values for Routing Problems, *Technical Report Series ORP99-05*, The University of Texas at Austin, Graduate Program in Operations Research.

Colletti, B. & Barnes, J. W. (2000a). Local Search Structure in the Symmetric Traveling Salesperson Problem Under a General Class of Rearrangement Neighborhoods, *Applied Mathematical Letters* **14**(1): 105–108.

Colletti, B. & Barnes, J. W. (2001). Local Search Structure in the Symmetric Traveling Salesperson Problem under a General Class of Rearrangement Neighborhoods, *Applied Mathematical Letters* **14**(99-06): 105–108.

Colletti, B. W. (1999). *Group Theory and Metaheuristics*, PhD thesis, The University of Texas, Austin, TX.

Colletti, B. W. & Barnes, J. W. (2000b). Linearity in the Traveling Salesman Problem, *Applied Mathematics Letters* **13**(3): 27–32.

Colorni, A., Dorigo, M., Maniezzo, V. & Trubian, M. (1994). Ant-System for Job-Shop Scheduling, *Belgian Journal of Operations Research, Statistics and Computer Science* **34**(1): 39–54.

Combs, T. (2001). A Group Theoretic Tabu Search Approach to Tanker Crew Scheduling, *Technical Report AFIT/DS/ENS/02*, Air Force Institute of Technology.

Cullenbine, C. A. (2000). *A Tabu Search Approach to the Weapons Assignment Model*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Dahl, G., Martin, A. & Stoer, M. (1999). Routing Through Virtual Paths in Layered Telecommunication Networks, *Operations Research* **47**(5): 693–702.

Dantzig, G. & Ramser, J. (1959). The Truck Dispatching Problem, *Management Science* **6**: 80–91.

Deitel, H. M. & Deitel, P. J. (1999). *Java How to Program*, third edn, Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458.

Della Croce, D., Tadei, R. & Rolando, R. (1994). Solving a Real World Project Scheduling Problem with a Genetic Approach, *Belgian Journal of Operations Research, Statistics and Computer Science* **33**(1/2): 65–78.

Dell'Amico, M. & Toth, P. (2000). Algorithms and Codes for Dense Assignment Problems: The State of the Art, *Discrete Applied Mathematics* **100**: 17–48.

Dell'Amico, M. & Trubian, M. (1993). Applying Tabu Search to the Job-Shop Scheduling Problem, *Annals of Operations Research* **41**: 231–252.

Derigs, U. (1985). The Shortest Augmenting Path Method for Solving Assignment Problems - Motivation and Computational Experience, *in* C. Monma (ed.), *Algorithms and Software for Optimization - Part I, Ann. Oper. Res., Vol.*, Baltzer, Basel, pp. 57–102.

Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M. M. & Soumis, F. (1997). Daily Aircraft Routing and Scheduling, *Management Science* **43**(6): 841–855.

Desrochers, M., Desrosiers, J. & Solomon, M. (1992). A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows, *Operations Research* **40**: 342–353.

Desrosiers, J., Sauve, M. & Soumis, F. (1988). Lagrangian Relaxation Methods for Solving the Minimum Fleet Size Multiple Traveling Salesman Problem with Time Windows, *Management Science* **34**(8): 1005–1022.

Erkut, E., Myroon, T. & Strangway, K. (2000). TransAlta Redesigns its Service-Delivery Network, *Interfaces* **30**(2): 54–69.

Ernst, A. T. & Krishnamoorthy, M. (1998). Exact and Heuristic Algorithms for the Uncapacitated Multiple Allocation P-Hub Median Problem, *European Journal of Operational Research* **104**: 100–112.

Fisher, M. L., Greenfield, A. J., Jaikumar, R. & Kedia, P. (1981). Real-Time Scheduling of a Bulk Delivery Fleet: Practical Application of Lagrangian Relaxation.

Fisher, M. & Rinnooy Kan, A. (1988). The Design, Analysis and Implementation of Heuristics (Special Issue), *Management Science* **34**(3): 263–401.

French, S. (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Halsted Press (John Wiley & Sons), New York.

Frizzell, P. W. & Giffin, J. W. (1995). The Split Delivery Vehicle Scheduling Problem with Time Windows and Grid Network Distances, *Computers & Operations Research* **22**(6): 655–667.

Gaglione, A. M. (1992). An Introduction to Group Theory, *Technical Report Report PU/NRL/5350–92-231*, Naval Research Laboratory, Washington, D. C.

Garcia, B. L., Potvin, J. Y. & Rousseau, J. M. (1994). A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints, *Computers & Operations Research* **21**: 1025–1033.

Garey, M., Johnson, D. & Sethi, R. (1976). The Complexity of Flowshop and Jobshop Scheduling, *Mathematics of Operations Research* **1**: 117–129.

Garey, M. R. & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman.

Gass, S. I. (1988). Preface, *in* B. L. Golden & A. A. Assad (eds), *Vehicle Routing: Methods and Studies*, North-Holland, New York, N.Y., chapter Preface, pp. v–vi.

Gendreau, M. & Laporte, G. (1996). A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers, *Operations Research* **44**: 469–477.

Gendreau, M., Laporte, G., Musaraganyi, C. & Taillard, E. D. (1999). A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem, *Computers & Operations Research* **26**: 1153–1173.

Gendreau, M., Laporte, G. & Potvin, G. (1997). *Vehicle Routing: Modern Heuristics*, John Wiley & Sons, Chichester, chapter 9, pp. 311–336.

Gilmore, P. C., Lawler, E. L. & Shmoys, D. B. (1985). *Well-Solved Special Cases*, John Wiley & Sons, New York, chapter The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, pp. 87–143.

Glover, F. (1969). Integer Programming Over a Finite Additive Group, *SIAM Journal on Control* **7**(2): 213–231.

Glover, F. (1989). Tabu Search - Part I, *ORSA Journal on Computing* **1**(3): 190–206.

Glover, F. (1995). Tabu Thresholding: Improved Search by Nonmonotonic Trajectories, *ORSA Journal on Computing* **7**(4): 426–442.

Glover, F., Kochenberger, G. A. & Alidaee, B. (1998). Adaptive Memory Tabu Search for Binary Quadratic Programs, *Management Science* **44**(3): 336–345.

Glover, F. & Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Boston, MA.

Golden, B. L. & Assad, A. A. (1988). *Vehicle Routing: Methods and Studies*, Vol. 16 of *Studies in Management Science and Systems*, North-Holland, New York, N.Y.

Gomory, R. E. (1958). Outline of an Algorithm for Integer Solutions to Linear Programs, *Bulletin of the American Mathematical Society* **64**: 275–278.

Gomory, R. E. (1963). *An Algorithm for Integer Solutions to Linear Programs*, McGraw-Hill, pp. 269–302.

Gomory, R. E. (1965). On the Relation Between Integer and Noninteger Solutions to Linear Programs, *Proceedings of the National Academy of Sciences USA* **53**: 260–265.

Gomory, R. E. (1969). Some Polyhedra Related to Combinatorial Problems, *Linear Algebra and its Applications* **2**(4): 451–558.

Gulf War Air Power Survey (1993). Vol. III, Logistics and Support, Washington, D. C.

Hall, S. N. (2000). *A Group Theoretic Tabu Search Approach to the Traveling Salesman Problem*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Harder, R. W. (2000). *A Jump Search Enhanced Reactive Tabu Search in a Universal Vehicle Router Applied to Routing Unmanned Aerial Vehicles*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Hart, J. P. & Shogan, A. W. (1987). Semi-Greedy Heuristics: An Empirical Study, *Operations Research Letters* **6**(3): 107–114.

Hillier, F. S. & Lieberman, G. J. (2001). *Introduction to Operations Research*, seventh edn, McGraw-Hill, New York.

Homberger, J. & Gehring, H. (1999). Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows, *Infor* **37**(3): 297–318.

Hostler, H. (1987). *Air Refueling Tanker Scheduling*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Isaacs, I. M. (1994). *Algebra*, Brooks/Cole Publishing, Pacific Grove, CA.

Jackson, J. (1956). An Extension of Johnson's Result on Job Lot Scheduling, *Naval Research Logistics Quarterly* **3**(3): 201–203.

Jacobs, L. & Brusco, M. (1995). A Local Search Heuristic for Large Set-Covering Problems, *Naval Research Logistics* **52**: 1129–1140.

Jain, A. & Meeran, S. (1999). Deterministic Job-Shop Scheduling: Past, Present and Future, *European Journal of Operational Research* **113**: 390–434.

Jarrah, A. I. & Strehler, J. C. (2000). An Optimization Model for Assigning Through Flights, *IIE Transactions* **32**: 237–244.

Jensen, P. A. & Barnes, J. W. (1980). *Network Flow Programming*, John Wiley & Sons, Inc., New York.

Johnson, E. L. (1980). *Integer Programming: Facets, Subadditivity, and Duality for Group and Semi-Group Problems*, Vol. 32 of *CBMS-NSF Regional Conference Series in Applied Mathematics.*

Johnson, S. (1954). Optimal Two- and Three-Stage Production Schedules and Set-Up Times Included, *Naval Research Logistics Quarterly* **1**: 61–68.

Junger, M. & Mutzel, P. (1997). 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms, *Journal of Graph Algorithms and Applications* **1**(1): 1–25.

Karp, R. M. (1979). A Patching Algorithm for the Nonsymmetric Traveling Salesman Problem, *SIAM Journal of Computing* **8**(4): 561–573.

Kelly, J. P. & Xu, J. (1999). A Set-Partitioning-Based Heuristic for the Vehicle Routing Problem, *INFORMS Journal on Computing* **11**(2): 161–172.

Killingsworth, P. S. (1996). *Multipoint Aerial Refueling: A Review and Assessment*, RAND, Santa Monica, CA.

Kinney, G. (2000). *A Hybrid Jump Search and Tabu Search Metaheuristic for the Unmanned Aerial Vehicle (UAV) Routing Problem*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Kohl, N. & Madsen, O. B. G. (1997). An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation, *Operations Research* **45**(3): 395–406.

Kontoravdis, G. & Bard, J. F. (1995). A GRASP for the Vehicle Routing Problem with Time Windows, *ORSA Journal on Computing* **7**: 10–23.

Kuhn, H. (1955). The Hungarian Method for the Assignment Problem, *Naval Res. Logistics Quart.* **2**: 83–97.

Kuhn, H. (1956). Variants of The Hungarian Method for the Assignment Problem, *Naval Res. Logistics Quart.* **3**: 253–258.

Laguna, M. & Glover, F. (1993). Integrating Target Analysis and Tabu Search for Improved Scheduling Systems, *Expert Systems With Applications* **6**: 287–297.

Laguna, M., Marti, R. & Campos, V. (1999). Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem, *Computers & Operations Research* **26**: 1217–1230.

Lahme, B. & Miranda, R. (1999). Karhunen-Loeve Decomposition in the Presence of Symmetry - Part I, *IEEE Transactions on Image Processing* **8**(9): 1183–1190.

Lawler, E. (1976). *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York.

Lederer, P. J. & Nambimadom, R. S. (1998). Airline Network Design, *Operations Research* **46**(6): 785–804.

Lenstra, J. K. & Rinnooy Kan, A. H. G. (1981). Complexity of Vehicle Routing and Scheduling Problems, *Networks* **11**: 221–227.

Løkketangen, A. & Glover, F. (1998). Solving Zero-One Mixed Integer Programming Problems Using Tabu Search, *European Journal of Operational Research* **106**: 624–658.

Lysgaard, J. (1992). Dynamic Transportation Networks in Vehicle Routing and Scheduling, *Interfaces* **22**(3): 45–55.

Marianov, V., Serra, D. & ReVelle, C. (1999). Location of Hubs in a Competitive Environment, *European Journal of Operational Research* **114**(2): 363–371.

Marin, A. & Salmeron, J. (1996). Tactical Design of Rail Freight Networks - Part I: Exact and Heuristic Methods, *European Journal of Operational Research* **90**(1): 26–44.

Marti, R. (1998). A Tabu Search Algorithm for the Bipartite Drawing Problem, *European Journal of Operational Research* **106**: 558–569.

Matsuo, H., Suh, C. & Sullivan, R. (1988). A Controlled Search Simulated Annealing Method for the General Job-Shop Scheduling Problem. Working paper, 03-04-88.

Moncrief, K. W. (1996). Tactical Air Refueling: Undocumented Past, Future Requirement, *Technical Report Graduate Research Paper AFIT/GMO/LAC/96N-9*, Air Force Institute of Technology.

Nanry, W. B. (1998). *Solving the Precedence Constrained Vehicle Routing Problem with Time Windows Using the Reactive Tabu Search Metastrategy*, PhD thesis, The University of Texas at Austin.

Nanry, W. P. & Barnes, J. W. (2000). Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search, *Transportation Research Part B* **34**: 107–121.

Nash, S. G. & Sofer, A. (1996). *Linear and Nonlinear Programming*, McGraw-Hill, New York.

Nembhard, D. A. & Chelsea C. White, I. (1999). A Heuristic Search Approach for Solving Multiobjective Non-Order-Preserving Path Selection Problems, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **29**(5): 450–459.

Nemhauser, G. L. & Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*, John Wiley & Sons, New York.

Neuway, D. L. (2001). Conjugative Neighborhoods: A Primer, *Technical Report Series ORP01-02*, The University of Texas at Austin, Graduate Program in Operations Research.
(`http://www.me.utexas.edu/~orie/techrep.html`).

Nowicki, E. & Smutnicki, C. (1996). A Fast Taboo Search Algorithm for the Job-Shop Problem, *Management Science* **42**(6): 797–813.

Ong, H. L., Ang, B. W., Goh, T. N. & Deng, C. C. (1997). A Vehicle Routing and Scheduling Problem with Time Windows and Stochastic Demand Constraints, *Asia-Pacific Journal of Operational Research* **14**(1): 1–17.

O'Rourke, K. P. (1999). *Dynamic Unmanned Aerial Vehicle (UAV) Routing with a Java-Encoded Reactive Tabu Search Metaheuristic*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Papadimitriou, C. H. & Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., Englewoods Cliffs, New Jersey 07632.

Pesch, E. (1993). Machine Learning by Schedule Decomposition. Working Paper.

Rego, C. (1998). A Subpath Ejection Method for the Vehicle Routing Problem, *Management Science* **44**(10): 1447–1459.

Renaud, J., Laporte, G. & Boctor, F. F. (1996). A Tabu Search Heuristic for the Multi-Depot Vehicle Routing Problem, *Computers & Operations Research* **23**: 229–235.

Resende, M. (1997). A GRASP for Job Shop Scheduling, San Diego, CA.

Romaine, J. M. (1999). *Solving the Multidimensional Multiple Knapsack Problem with Packing Constraints Using Tabu Search*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Roy, B. & Sussmann, B. (1964). Les Problèms D'ordonnancement Avec Contraintes Disjonctives, *Note D.S. no. 9 bis, SEMA, Paris, France* .

Rugelj, J. & Klavzar, S. (1997). Distributed Multicast Routing in Point-to-Point Networks, *Computers & Operations Research* **24**(6): 521–527.

Russina, B. & Ruthsatz, B. (1999). The Quick Look Tool for Tanker Deployment, *Technical report*, Center for Optimization and Semantic Control, St. Louis, MO.

Ryan, J. L., Bailey, T. G., Moore, J. T. & Carlton, W. B. (1999). Unmanned Aerial Vehicle (UAV) Route Selection Using Reactive Tabu Search. To appear in *Military Operations Research*.

Ryer, D. (2001). Personal Communication.

Ryer, D. M. (1999). *Implementation of the Metaheuristic Tabu Search in Route Selection for Mobility Analysis Support System*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Salkin, H. M. (1975). *Integer Programming*, Addison-Wesley, Reading, Massachusetts.

Sancho, N. G. F. (1997). The Hierarchical Network Design Problem with Multiple Primary Paths, *European Journal of Operational Research* **96**(2): 323–328.

Savelsbergh, M. W. P. (1985). Local Search in Routing Problems with Time Windows, *Annals of Operations Research* **14**: 285–305.

Savelsbergh, M. W. P. (1992). Computer Aided Routing, *Technical Report CWI Tract 75*, CWI.

Savelsbergh, M. W. P. & Goetschalckx, M. (1995). A Comparison of the Efficiency of Fixed versus Variable Vehicle Routes, *Journal of Business Logistics* **16**(1): 163–187.

Sherali, H. D. & Tuncbilek, C. H. (1997). Static and Dynamic Time-Space Strategic Models and Algorithms for Multilevel Rail-Car Fleet Management, *Management Science* **43**(2): 235–250.

Sisson, M. R. (2000). *Applying Tabu Heuristic To Wind Influenced, Minimum Risk, and Maximum Expected Coverage Routes*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Smith, R. K. (1998). *75 Years of Inflight Refueling Highlights, 1923-1998*, U.S. Government Printing Office, Washington, D.C.

Solomon, M. & Desrosiers, J. (1988). Survey Paper: Time Window Constrained Routing and Scheduling Problems, *Transportation Science* **22**(1): 1–13.

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research* **35**(2): 254–265.

Strategic Air Command (1990). *Seventy Years of Strategic Air Refueling, 1918-1988: A Chronology*, Office of the Historian, Headquarters Strategic Air Command, Offutt AFB, NE.

Subramanian, R., Richard P. Scheff, J., Quillinan, J. D., Wiper, D. S. & Marsten, R. E. (1994). Coldstart: Fleet Assignment at Delta Air Lines, *Interfaces* **24**: 104–120.

Suh, C. (1988). *Controlled Search Simulated Annealing for Job Scheduling*, PhD thesis, The University of Texas at Austin.

Sunderman, J. F. (1961). *Early Air Pioneers*, Franklin Watts, Inc., New York, NY.

Taillard, È. (1989). Parallel Taboo Search Techniques for the Job-Shop Scheduling Problem, *Technical Report Internal Research Report ORWP89/11*, Dèpartment de Mathèmatiques (DMA), Ècole Polytechnique Fèdèrale de Lausanne, 1015 Lausanne, Switzerland.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F. & Potvin, J. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Windows, *Transportation Science* **31**: 170–186.

Tamm, U. (1998). Splittings of Cyclic Groups and Perfect Shift Codes, *IEEE Transactions on Information Theory* **44**(5): 2003–2009.

Thangiah, S. R., Potvin, J.-Y. & Sun, T. (1996). Heuristic Approaches to Vehicle Routing with Backhauls and Time Windows, *Computers & Operations Research* **23**(11): 1043–1057.

Van Laarhoven, P., Aarts, E. & Lenstra, J. (1992). Job Shop Scheduling by Simulated Annealing, *Operations Research* **40**(1): 113–125.

Werner, F. & Winkler, A. (1995). Insertion Techniques for the Heuristic Solution of the Job-Shop Problem, *Discrete Appl. Math.* **58**(2): 191–211.

White, W. W. (1966). *On a Group Theoretic Approach to Linear Integer Programming*, PhD thesis, University of California at Berkeley.

Wiley, V. D. (2000a). A Step-by-Step User's Guide to Implementating the Group Class in Java, *Technical Report Series ORP00-05*, The University of Texas at Austin, Graduate Program in Operations Research. (`http://www.me.utexas.edu/~orie/techrep.html`).

Wiley, V. D. (2000b). The SymmetricGroup Class User's Manual for Partitioning and Ordering Problems, *Technical Report Series ORP00-04*, The University of Texas at Austin, Graduate Program in Operations Research. (`http://www.me.utexas.edu/~orie/techrep.html`).

Winston, W. L. (1994). *Operations Research, Applications and Algorithms*, third edn, Duxbury Press, Belmont, California.

Wolsey, L. A. (1969). *Mixed Integer Programming: Discretization and the Group Theoretic Approach*, PhD thesis, Massachusetts Institute of Technology.

Wolsey, L. A. (1998). *Integer Programming*, John Wiley & Sons, New York.

Xu, J., Chiu, S. Y. & Glover, F. (1999). Optimizing a Ring-Based Private Line Telecommunication Network Using Tabu Search, *Management Science* **45**(3): 330–345.

Xu, J. & Kelly, J. (1996). A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem, *Transportation Science* **30**: 379–393.

Yamaha, T. & Nakano, R. (1996a). *Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search*, Kluwer Academic Publishers, Hingham, MA, chapter Meta-heuristics; Theory and Applications, pp. 237–248.

Yamaha, T. & Nakano, R. (1996b). Scheduling by Genetic Local Search with Multi-Step Crossover, *Fourth International Conference on Parallel Problem Solving from Nature (PPSN IV)*, Berlin, Germany, pp. 960–969.

Yamani, A. A. (1986). *Analysis of an Air Transportation System*, PhD thesis, University of Florida.

Yan, S. & Tu, Y.-P. (1997). Multifleet Routing and Multistop Flight Scheduling for Schedule Perturbation, *European Journal of Operational Research* **103**(1): 155–169.

Zeisler, N. J. (1999). *A Reactive Tabu Approach to Solving Air Mobility Command's Intratheater Airlift Problem*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
20/Sep/2001	MAJOR REPORT	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
PATENTS VS PATIENTS	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
MAJ REINHOLD HERMAN	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
JA GENERAL SCHOOL ARMY	CI01-184

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
THE DEPARTMENT OF THE AIR FORCE AFIT/CIA, BLDG 125 2950 P STREET WPAFB OH 45433	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
				16	19b. TELEPHONE NUMBER *(Include area code)*

-SEARCHING FOR HEROES
Book Review of *THE TERRIBLE HOURS*
By Major Herman Reinhold*

* Judge Advocate, United States Air Force. A.A., 1979, Columbia-Greene Community College, Hudson, New York. B.A., 1981, State University of New York at Albany. J.D., 1984, School of Law, State University of New York at Buffalo. LLM expected May 2001, The Judge Advocate General's School, Charlottesville, Virginia. Currently assigned as a LL.M. student at the Judge Advocate General's School, Charlottesville, Virginia. Formerly assigned to: 15th Air Force, Travis Air Force Base (AFB) CA, (Chief of General Law) 1997-2000; 319th Aerial Refueling Wing, Grand Forks AFB (Deputy Staff Judge Advocate) 1993-1997; Air Force Defense Test Center, Eglin AFB (Chief, Military Justice) 1991-1993; 487th Tactical Missile Wing, Comiso Air Station (Deputy Staff Judge Advocate) 1990-1991; Area Defense Counsel, Griffiss AFB (1989-1990); 416th Bombardment Wing, Griffiss AFB (Chief, Civil Law) 1987-1989). This article is a book review that was submitted in partial completion of the Master of Laws requirement of the 49th Judge Advocate Officer Graduate Course. Major Reinhold is a member of the New York bar and has been admitted to practice before the Court of Appeals of the Armed Forces.

When the Russian submarine *Kursk* sank, could one man have saved the crew? Author Peter Maas thinks the answer is "YES!" In August 2000, when the *Kursk* sank, Maas had a sub book on the bestseller list.[1] His book, *The Terrible Hours: The Man Behind The Greatest Submarine Rescue In History*, tells of the rescue of the crew of the *USS Squalus* in 1939.[2] Because he knew the details of the *Squalus* rescue, Maas spoke with reporters on television news shows and was quoted in USA TODAY.[3] He had a simple message - the unsung hero of *The Terrible Hours*, Charles "Swede" Momsen, could have saved the Russian crew using his diving bell.[4] Maas said: "The difference is that the Russians didn't have a Swede Momsen."[5]

Maas wrote two books on Momsen and the *Squalus*. Maas first wrote of Momsen and the *Squalus* rescue in his 1967 book *The Rescuer*,[6] which was excerpted in the *SATURDAY EVENING POST*.[7] Interest in World War II history caused Maas to rewrite *The Rescuer* to create *The Terrible Hours*.[8]

---

[1] *Bestseller List*, USA TODAY, Aug 24, 2000 at 6D

[2] PETER MAAS, THE TERRIBLE HOURS: THE MAN BEHIND THE GREATEST SUBMARINE RESCUE IN HISTORY (HarperTorch, 2000) (1999)

[3] *RIVERA LIVE* (CNBC news show, Aug 22, 2000); *CNN TODAY* (CNN news show, Aug 15, 2000); *SUNDAY TODAY* (NBC news show, Oct 10, 2000) (television transcripts available through www.lexis.com); Bob Minzeheimer, *Two Sub Sinkings Have One Terrible Difference*, USA TODAY, Aug. 24, 2000, at 6D

[4] See *RIVERA LIVE, supra* note 3, *CNN TODAY, supra* note 3, *SUNDAY TODAY, supra* note 3, and Minzeheimer, *supra* note 3

[5] Minzeheimer, *supra* note 3

[6] PETER MAAS, THE RESCUER, (Harper, 1968)

[7] Peter Maas, *The Rescuer*, SATURDAY EVENING POST, Sept. 23, 1967, 36-69

[8] MAAS, *supra* note 2, 309, and Minzesheimer, *supra* note 3

Military professionals should read *The Terrible Hours*, but with a critical eye. Momsen was an impressive man who did many interesting and heroic things. Maas wrote an entertaining and informative book with many examples of heroism. However, *The Terrible Hours* is too subjective to give a fair picture of Momsen. The book is also limited by how Maas portrays Momsen as a hero. The book may even harm Momsen's reputation among readers, if they mistakenly see Momsen as a man eager to take credit for the work of others.

Does Momsen's reputation prove him to be a hero? What is a hero? A hero faces strong opposition and does his or her very best, thereby earning admiration for bravery or courage. A person may be a "hero" for doing one or many bold acts, and people often focus on those specific noteworthy acts.

Maas is convinced Momsen is a hero and gives many specific examples to convince readers. Momsen flunked out of Annapolis, fought for a second appointment, returned and graduated. He was a submarine captain and saved his ship and crew when it was trapped on the ocean floor. He bravely tested new ways to save sub crews. He fought for his ideas even when opposed by superior officers. Momsen developed and tested new ways for men to dive. He saved the survivors of the *Squalus* and then raised the sunken sub to be studied and salvaged.[9]

After the *Squalus* salvage, Momsen continued his distinguished Navy career. He was at Pearl Harbor and reacted quickly to reports of mini-subs, ordering destroyers to search for the subs. During World War II, Momsen fixed serious problems with the Navy mail system, torpedoes, and explosive powder that ignited spontaneously. Momsen developed and tested new attack strategies for subs. He was captain of the battleship *USS South Dakota*. After the war, Momsen helped design a prototype submarine for the Navy, the *USS Albacore*.[10]

While Momsen was a hero, Maas omits many great heroic events from Momsen's 36 year Navy career. In *The Rescuer*, Maas reports that after World War II, Momsen safely returned 5,700,000 Japanese colonizers to Japan. General Douglas MacArthur praised him for this accomplishment.[11] Momsen's 1945-1951 return of colonizers was when he ran Japan's merchant marine. Maas does not report the return in *The Terrible Hours*. Mass also omits that Momsen commanded: the Submarine Force, Pacific Fleet; the First Naval District; and, Joint Task Force Seven.[12]

Because of Momsen's many accomplishments, Maas should have written a traditional biography instead of a rescue story. Maas is best known for his nonfiction biographies *The Valachi Papers*, *Serpico* and *King Of The Gypsies*.[13] These books each focus on an individual hero, struggling alone against organized crime, police corruption or deep family problems.[14]

---

[9] MAAS, *supra* note 2, at 32-36, 63-77, 72-77, 116-132, 147-165, 177-309

[10] MAAS, *supra* note 2, at 291-309

[11] MAAS, THE RESCUER, *supra* note 5, at 218-219

[12] *Biography of Vice Admiral Charles B. Momsen, USN, (Retired) (1896-1967)*, Department of the Navy, Naval Historical Center, available at http://www.history.navy.mil/photos/per-us/uspers-m/cb-momsn.htm (last visited Sept. 15, 2000).

[13] PETER MAAS, THE VALACCHI PAPERS (Putnam, 1969); PETER MAAS, SERPICO, (Viking, 1973); PETER MAAS, KING OF THE GYPSIES (Viking, 1975); BIOGRAPHY OF PETER MAAS, (Contemporary Authors on CD-ROM, Gale Research, 1998), 1-2

[14] BIOGRAPHY OF PETER MAAS, (Contemporary Authors on CD-ROM, Gale Research, 1998), *supra* note 13, at 3

Similarly, in *The Terrible Hours*, Maas shows Momsen bravely standing alone against Navy bureaucracy or the forces of nature. Maas makes it his duty or even obsession to get Momsen the proper recognition.[15] Maas' strategy is exaggeration of Momsen's role and successes. Maas focuses on Momsen, discounts the work of other people, and ignores the historical context of Momsen's work. Interestingly, as explained below, Momsen is often more willing than Maas to share the credit for Momsen's work.

Momsen's work to save the *Squalus* crew is central to the book. Maas uses Momsen's development of rescue techniques as part of the background of the rescue. Maas writes:

> Everything that could possibly save a trapped submariner – smoke bombs, telephone marker buoys, new deep-sea diving techniques, escape hatches and artificial lungs, a great pear-shaped diving bell, or rescue chamber – was either a direct result of his inventive, pioneering derring –do, or of value only because of it.[16]

This ignores the facts. Momsen was on teams that developed escape hatches, artificial lungs, and a rescue chamber.[17] A month after the *Squalus* was returned to port, Momsen gave credit to the Navy when he said:

> Those were, briefly, the high lights (sic) of the concluding chapter of the story of 12 years of research and training by the Navy after the tragic loss of the S-4 (SS-109) in 1927. In the 33 survivors, the Navy paid a "dividend" on the time and money spent in preparedness.[18]

Maas also shows his bias towards Momsen when discussing rescue equipment. Momsen's first great achievement in the development of rescue technology was the Momsen Lung, which allowed submariners to breathe normally when swimming to the surface from a sunken sub. The lung was developed in 1929-1932, but Maas does not give the dates.[19] Maas calls the lung "a completely fresh approach to saving submariners" and "a daring new concept."[20] But, Maas also hints at other devices invented before the lung by saying they were "too bulky or balky" and there was no "serious effort" to train sub crews in their use.[21] Maas does not mention any navy

---

[15] *SUNDAY TODAY supra* note 3

[16] MAAS, *supra* note 2, at 33

[17] MAAS, *supra* note 2, at 73-77, 116-132, 152-165

[18] Charles Momsen, *Lecture by Charles Momsen on Rescue and Salvage of USS Squalus, Delivered to Harvard Engineering Society, Oct 6, 1930*, Department of the Navy, Naval Historical Center, available at http://www.hstory.navy.mil/faqs/faq99-6.htn, (last visited Sept. 15, 2000)

[19] *Biography of Vice Admiral Charles B. Momsen, USN (Retired) (1896-1967)*, *supra* note 12

[20] MAAS, *supra* note 2, at 116

[21] MAAS, *supra* note 2, at 117

adopting a device before the U. S. Navy adopted the Momsen Lung. However, the German Navy adopted a similar device in 1912, and the British Navy did so in the 1920's.[22]

Maas did not discuss the history of the Momsen Lung after the *Squalus* rescue, when the crew held them as a last possible resort in case the rescue bell did not work. In the only known use of the lung during World War II, eight men used lungs to escape from the *USS Tang* when it sank in 180 feet of water in October 1944.[23] After World War II, new procedures allowed escape from 300 feet without any breathing devices. It has even been argued that the Momsen Lung cost lives, by giving submariners the false impression they needed a device to leave a sunken ship.[24]

Maas ignores even more history when he discusses the McCann Rescue Chamber; a large, modern, diving bell designed to save trapped submariners. Diving bells have a long history. Since Aristotle first wrote about diving bells, they were repeatedly improved over the centuries as they were used for underwater exploration and work.[25] By 1792, diving bells had air pumps to deliver fresh air from the surface.[26] Momsen's genius was in helping design, develop and test a bell especially made to save submarine crews.[27]

Maas focuses on how the chamber was named. According to Maas, the US Navy named it after Lieutenant Commander Allen McCann, who merely tested the bell; it was named after McCann because Momsen had "stepped on too many toes" in his work to save submariners.[28] However, Momsen is more willing to give McCann credit. In October 1939, at a lecture on the *Squalus* rescue and salvage, he said: "My memory went back to...the first diving bell, the cranky open bell that would dump and fall and half drown us if we were not careful, of the final design produced by Commander Allen R. McCann and the comfort that it was to operate."[29] Momsen, speaking with first hand knowledge in 1939, is more gracious about McCann's contributions than Maas.

Maas also fails to give fair credit for the *Squalus* rescue and recovery. Momsen was part of a large team that included navy commanders on land and sea, several ships and crews, and teams of divers.[30] Maas gives the bulk of the credit to Momsen. Maas also explains why he feels Momsen deserves credit. Some people disagree.

Admiral C. W. Cole commanded the rescue of the *Squalus* and called both Momsen and McCann to the scene. In his final report, he lists his crew and says McCann was a "Technical

---

[22] THE ENCYCLOPEDIA AMERICANA INTERNATIONAL EDITION (Grolier Incorporated, 1996), Submarine, 820-821

[23] CLAY BLAIR, JR., SILENT VICTORY: THE U. S. SUBMARINE WAR AGAINST JAPAN (J. B. Lippincott + Company, 1975), 767-769

[24] *Id.* 768 footnote

[25] THE NEW ENCYCLOPEDIA BRITANNICA, (Encyclopedia Britannica, 1998), Diving Bell, Volume 4, 133 (Encyclopedia Britannica, 1998)

[26] *Id.* Smeaton, John, Volume 10, 889

[27] MAAS, *supra* note 2, at 73-77, 122-128, 155-164

[28] MAAS, *supra* note 2, at 163-164

[29] Charles Momsen, *supra* note 18

[30] EDWARD P. STAFFORD, THE FAR AND THE DEEP (G. P. Putnam's Sons, 1966), 124-134

Aide" and Momsen was the "Diving Officer." Momsen supervised the divers but was too old to dive himself. In the conclusion of his report, Cole "invites" attention to the:

> [F]ollowing, which are deemed worthy of the highest praise:...The efficient work of the divers...The exceptional coolness, judgment and initiative of Commander Allan R. McCann in handling what was probably the most trying and difficult situation of the rescue period, viz: the fourth and last trip up of the rescue chamber with survivors.[31]

Cole does not highlight Momsen by name for any special praise in his report.

A 1942 book by David O. Woodbury gives McCann the credit for developing the bell and for being the hero of the *Squalus* rescue effort.[32] Woodbury reports Momsen at the rescue but does not mention any specific acts by Momsen.[33]

Commander Edward Peary Stafford did not consider either Momsen or McCann the hero of the rescue. In his 1966 book, Stafford gives the credit to Admiral Cole, saying his "instant, vigorous and appropriate action resulted in the rescue of every live man in the sunken *Squalus*."[34] Selecting Cole as the hero makes sense, perhaps because he brought the other two "heroes" or "supporting players," McCann and Momsen, to the scene. Other possible heroes include the *Squalus*' captain and crew, and the many divers who rescued the crew and salvaged the ship.

Focusing on selecting "a hero" misses the point. The rescue and salvage were successes of a Navy team, built on the hard work of men who went before them. While Momsen's work was important, he was just one man. He could not, and did not, develop the equipment or procedures alone. He could not save the *Squalus* crew or salvage the ship by himself, nor could he have saved the crew of the *Kursk* alone, had he been alive.

Readers of *The Terrible Hours* may wonder why Maas worked so hard to promote Momsen as a hero. The bias Maas showed for Momsen, and against giving others due credit, hurts Maas' credibility. Was Maas disappointed that McCann and Cole got more credit than Momsen? Was Maas impressed with Momsen after meeting and interviewing him? Was Maas' bias because he had more information about Momsen? Momsen died of cancer in 1967, just before *The Rescuer* was published. Was Maas sympathetic to Momsen after Momsen's death? Was Maas writing an extended obituary for Momsen?

Maas met Momsen late in Momsen's life. Maas, a U.S. Navy journalist from 1952 to 1954, was assigned to write about the *USS Albacore*, and in doing so learned about Momsen and the *Squalus*. After leaving the Navy, Maas met and interviewed Momsen and was given access to

---

[31] Rear Admiral C. W. Cole, *Report of Commander Rescue Operations, USS Squalus, May 28, 1939*, Department of the Navy, Naval Historical Center, available at http://www.hstory.navy.mil/faqs/faq99-3.htn (last visited Sept. 15, 2000)

[32] DAVID O. WOODBURY, WHAT THE CITIZEN SHOULD KNOW ABOUT SUBMARINE WARFARE (W.W. Norton & Company, 1942), 140-157

[33] *Id.* 147-157

[34] STAFFORD, *supra* note 30, 162, photo caption between 192 and 193

Momsen's personal papers. Maas interviewed people who knew Momsen and interviewed many of the *Squalus* survivors. Maas also checked newspaper accounts of the disaster.[35]

Maas turned his research into *The Rescuer*, his first full-length book.[36] It is a 239-page book with many advantages over the 309 pages of *The Terrible Hours*. *The Rescuer* has footnotes, an index, a bibliography, diagrams (of the submarine, the rescue bell and the pontoons used to salvage the sub) and twelve pictures of the rescue operation and of Momsen. The excerpt of *The Rescuer* printed in the *SATURDAY EVENING POST* also had pictures and a diagram.[37] The pictures and diagrams help explain the events. *The Terrible Hours* has no pictures, diagrams, footnotes, index or bibliography. When comparing the books, minor discrepancies appear in the quoted dialogue and in the reported events. Maas does not explain these differences. Without footnotes or a bibliography, it is difficult to know which book is more accurate or to cross-reference his report of the events with other sources.

When Maas wrote *The Rescuer* and *The Terrible Hours*, he missed a great chance to write a biography. Momsen deserves a complete, objective biography that lets readers learn from his successes and mistakes. Even after reading both of Maas' books, readers may want more information. Momsen did great things in war and in peace, even when superiors discounted his ideas. Momsen succeeded while working with technology and within a bureaucracy. What made him such a good leader and so effective? Maas does not explore Momsen's long, consistent record of innovation, leadership, consistency and dedication; which is his strongest legacy. The *Squalus* rescue was just one event in a long successful career. If someone feels Momsen is a hero because of the *Squalus* rescue, he or she may be disappointed to learn of the great contributions of others, who may be as much or more of a hero.

Hero worship is a limited field, and frankly, the military needs leaders more than heroes. A movie like RAMBO unrealistically shows the superhuman "bullet-proof" strength of the "hero." A "hero" may inspire some people around him to accomplish great feats. However, a film like APOLLO 13 shows people working together as a team to accomplish something great. A great leader will inspire followers to great accomplishments through teamwork. The team is often greater than the sum of its parts. In today's world of technology and bureaucracy, we need accomplished leaders. Heroes are not as important as leaders who can make large groups effective. Momsen was more than just a hero.

In the final analysis, Maas was effective, because readers will remember Momsen. However, readers must take this sea tale with a grain of salt. *The Terrible Hours* will help people remember Charles "Swede" Momsen. However, the best way to preserve his name and memory would have been with a balanced and detailed biography. Perhaps *The Terrible Hours* will inspire a historian to write a biography of Momsen. That book should focus on his accomplishments, in context, and explain how Momsen was able to work on and lead so many successful teams to wonderful accomplishments. The real message of the *Kursk* disaster and the rescue of the *Squalus* is that we usually succeed or fail, win or lose, live or die, as part of a team.

---

[35] MAAS, *supra* note 2, at 306-309 for Maas' communications with Momsen; and see BIOGRAPHY OF PETER MAAS, (Contemporary Authors on CD-ROM, Gale Research, 1998), *supra* note 13, at 1 for dates of Maas' military service

[36] MAAS, *supra* note 2, 302-309

[37] Maas, *supra* note 7, 36-40